AD-756 497

EFFECTS OF TERRAIN POWER SPECTRAL DENSITY SHAPING AND MEASUREMENT INTERVAL ON A VEHICLE RIDE SIMULATION

Robert E. Keenan, Jr.

Stevens Institute of Technology

Prepared for:

Department of Defense

February 1973

DISTRIBUTED BY:

NTIS

National Technical Information Service
U. S. DEPARTMENT OF COMMERCE
5285 Port Royal Road, Springfield Va. 22151

R-1646

DAVIDSON LABORATORY

Report SIT-DL-73-1646

February 1973

EFFECTS OF TERRAIN POWER SPECTRAL DENSITY SHAPING AND MEASUREMENT INTERVAL ON A VEHICLE RIDE SIMULATION

by

Robert E. Keenan, Jr.

prepared for
Department of Defense
under
Contract DAAE-07-69-0356
(Project THEMIS)

STEVENS INSTITUTE OF TECHNOLOGY
CASTLE POINT STATION
HOBOKEN, NEW JERSEY 07030

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Davidson Laboratory, Stevens Institute of Technology Hoboken, New Jersey 07030	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP

3. REPORT TITLE

EFFECTS OF TERRAIN POWER SPECTRAL DENSITY SHAPING
AND MEASUREMENT INTERVAL ON A VEHICLE RIDE SIMULATION

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Final Report

5. AUTHOR(S) (First name, middle initial, last name)

Robert E. Keenan, Jr.

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
February 1973	~~115~~ 126	8

8a. CONTRACT OR GRANT NO. DAAE-07-69-0356 b. PROJECT NO. c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10. DISTRIBUTION STATEMENT This document has been approved for public release and sale; its distribution is unlimited. [redacted] Reproduction of the document in whole or in part is permitted for any purpose of the U.S. Govt.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY Department of Defense Washington, D. C. 20301

13. ABSTRACT

A two-dimensional, five-degree-of-freedom, digital computerized, wheeled-vehicle-ride simulation is tested for sensitivity to two parameters: the power spectral density slope of computer-generated, random terrain profiles and the spacing of the discrete profile points. The vehicle ride simulation is exercised over six terrain profiles of different PSD slopes but identical RMS elevations. The simulation is also exercised several times over one basic profile described by samples taken at different measurement intervals. Calculated absorbed power at the vehicle center of gravity is used to compare ride roughness over the different profiles. The vehicle simulation is shown to be extremely sensitive to changes in PSD slope. The sensitivity to changes in measurement interval is shown to be dependent on vehicle size and mass. ( )

DD FORM 1473 (PAGE 1)

I-a

UNCLASSIFIED

UNCLASSIFIED

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
POWER SPECTRAL DENSITY						
RANDOM TERRAIN PROFILE						
VEHICLE RIDE SIMULATION						
SENSITIVITY ANALYSIS						
MEASUREMENT INTERVAL						

DAVIDSON LABORATORY
Stevens Institute of Technology
Castle Point Station
Hoboken, New Jersey 07030

Report SIT-DL-73-1646

February 1973

EFFECTS OF TERRAIN POWER SPECTRAL DENSITY SHAPING AND MEASUREMENT INTERVAL ON A VEHICLE RIDE SIMULATION

by

Robert E. Keenan, Jr.

Prepared for
Department of Defense
under
Contract DAAE-07-69-0356
(DL Project (3683/423)

Approved

I. Robert Ehrlich, Manager
Transportation Research Group

I-C

ABSTRACT

Effects of Terrain Power Spectral Density Shaping
and Measurement Interval on a Vehicle Ride Simulation

by

Robert E. Keenan, Jr.

Advisor

I. Robert Ehrlich

May 1972

A two-dimensional, five-degree-of-freedom, digital computerized, wheeled-vehicle-ride simulation is tested for sensitivity to two parameters: the power spectral density slope of computer-generated, random terrain profiles and the spacing of the discrete profile points. The vehicle ride simulation is exercised over six terrain profiles of different PSD slopes but identical RMS elevations. The simulation is also exercised several times over one basic profile described by samples taken at different measurement intervals. Calculated absorbed power at the vehicle center of gravity is used to compare ride roughness over the different profiles. The vehicle simulation is shown to be extremely sensitive to changes in PSD slope. The sensitivity to changes in measurement interval is shown to be dependent on vehicle size and mass.

KEY WORDS

Power Spectral Density

Random Terrain Profile

Vehicle Ride Simulation

Sensitivity Analysis

Measurement Interval

TABLE OF CONTENTS

LIST OF SYMBOLS

A_{ij}	smoothing coefficients	- -
C	constant	- -
C_{ij}	covariance, autocovariance	in
D	horizontal displacement	in
D'	vertical displacement	in
F_i	force on axle	lb
F_2'	force on rear bogie	lb
f	frequency	cycles/in
g	acceleration due to gravity	in/sec^2
h	positive integer	-
I_o	pitch moment of inertia	$lb\text{-}sec^2/in$
K	equivalent degrees of freedom	-
k	spring constant	lb/in
ℓ_i	geometrical vehicle parameters	in
m	maximum lag	-
m_i	mass	$lb\text{-}sec^2/in$
N	number of points in profile	-
n	slope of PSD curve	-
P	power spectral density estimate	in^2/cycle/in
P_i	ground force on axle	lb
r	positive integer	-
t	time (or distance)	sec (in)
U(r)	smoothed spectral density estimate	in^2/cycle/in

LIST OF SYMBOLS (Cont'd)

V	random normal number	-
V(r)	raw spectral density estimate	in^2/cycle/in
X	displacement	in
Y	displacement	in
z	uniform random number	-
z_i	vertical displacement	in

Greek Letters

α	spacial cutoff frequency	cycles/in
β	angle	radians
γ	vertical spring force component	lb/in
Δ	suspension displacement	in
$\Delta\ell$	profile point spacing	in
θ	pitch angle	radians
π	constant (3.14159...)	-
σ_n	standard deviation	-
τ	profile spacing in NOIPSD	in
	lag in literature on PSD	-

Other

<	less than
≤	less than or equal to
>	greater than
≥	greater than or equal to
→	approaches continuously
*	multiplication (eg. 2*3 = 6)

LIST OF FIGURES

LIST OF FIGURES (Con't)

LIST OF TABLES

INTRODUCTION

For some time, off-road mobility has been of manifest importance to designers of military vehicles. The problem has been to design a vehicle to traverse terrain with certain characteristics at a reasonable speed with negligible adverse effects on vehicle payload or driver.

This study concerns itself with some of the problems encountered in describing and measuring terrain characteristics -- in particular, ground roughness.

BACKGROUND

The characteristics of a terrain can be numerous. They may include, among others, slope, obstacles, vegetation, soil strength, and roughness. It has been shown, however, that the single most speed-limiting aspect of off-road mobility is ride dynamics.[1] The study of ride dynamics concerns itself with human and cargo response to vibration. This vibration may be caused by terrain roughness and/or discrete obstacles, and filtered by the vehicle mass, geometry, and suspension system. Vibration caused by traversal of discrete obstacles (logs, small ditches) is transient in nature and vehicle speed is limited more by vehicle strength and operator judgement and experience than by vibrational characteristics.[2] On the other hand, vibration due to stationary ground roughness is close to a steady-state condition; dependent on vehicle velocity. An operator will, if properly motivated, increase the vehicle speed until some degree of

discomfort is felt. He will then decrease speed slightly. The maximum speed a vehicle with driver can maintain over a certain terrain, then, is determined by:

(1) Terrain roughness

(2) Vehicle strength and suspension

(3) Driver discomfort (or payload delicacy)

It has long been recognized that some quantitative measure of the last of these, driver discomfort, is necessary to ride dynamics research. Many studies have been conducted in an attempt to determine some subjective measure. Van Deusen shows the apparent futility of this approach and some of the resultant confusion in a composite graph, reproduced in Figure 1.[2]

Several quantitative measures have been used with greater success. They include, but are not limited to: RMS acceleration at the driver's seat (or the area under the RMS acceleration time history)[1]; maximum acceleration at the driver seat[2]; the amplitude probability distribution of driver acceleration[1]; and absorbed power, a concept generated by Lee and Pradko[3] which is intended to quantify the energy dissipated by the human in the vibration of his limbs and flesh and counteracted partially by muscular control. In the words of Lee and Pradko[3]:

> The important characteristics of absorbed power are are that it has physical significance and therefore can be measured as well as computed analytically; and that since power is a scalar quantity, absorbed power can be summed in complex multidegree of freedom systems to determine human response.

The results of an earlier effort by the same authors show that absorbed power levels measured at the driver's seat seem to agree more

1. Ziegenruecker and Magid "Actual bodily harm feared"
2. Magid & Coermann "One minute tolerance"
3. Diekmann "Intolerable"
4. Gorrill & Snyder "Alarming"
5. Parks and Snyder "Alarming" (3/4" felt pad)
6. Zeller "Upper tolerance limit"
7. Goldman "Intolerable"

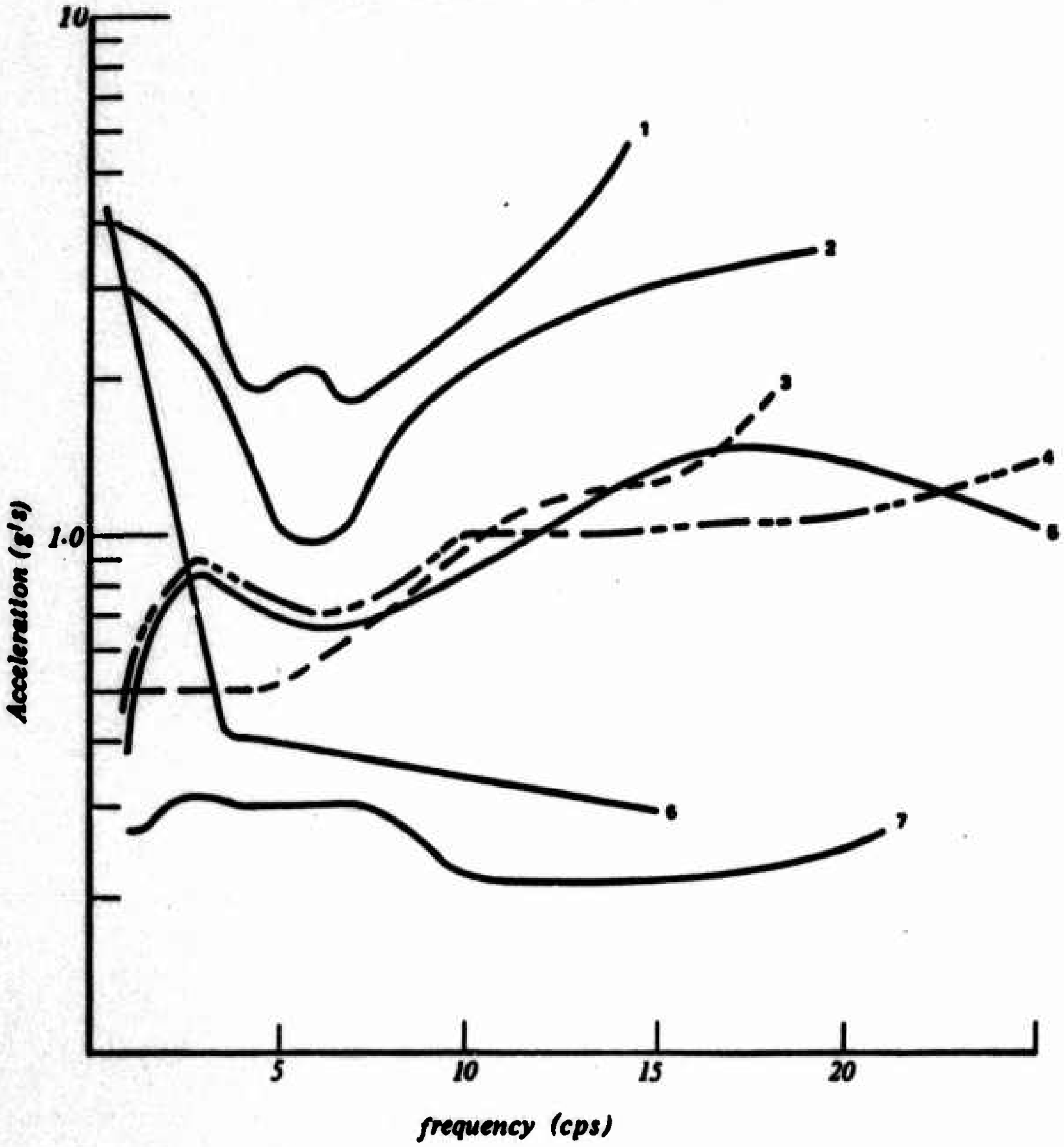


Figure 1. Subjective Ride Criterion

closely with subjective ride evaluations than do RMS accelerations.[4] Thus, the current study will use the absorbed power concept. Even among proponents of this criteria, there is confusion over the question of what level of absorbed power should be considered speed-limiting. Six watts, being a generally accepted figure, will be used herein with no apologies to those who consider it too low.

This research concerns itself, not with the driver discomfort or vehicle systems problem, but with the first factor mentioned above -- terrain roughness.

Recently,(1972)power spectral density (PSD) estimates have been used to describe terrain roughness statistically. The mathematical definitions of PSD will be discussed in some detail later. When PSD estimates for real terrain waveforms are required, the problems of aliasing and stationarity become important. Aliasing is a condition where false evidence of certain frequencies appears because the discrete sampling interval is longer than the shortest wavelength present. Stationarity, briefly, is the absence of long-range changes in underlying statistical properties.

The first proposal of the PSD's use to characterize stable ground roughness came from Kozin, Cote, and Bogdanoff in 1963.[5] They found that for "visually constant" ground roughness, stationarity could be safely assumed for terrain segments up to 2000 feet. They used running averages to correct for unavoidable long-range trends such as hills.

Others have shown that the PSD curves of most natural and man-made surfaces can be approximated with the equation:

$$PSD(f) = Cf^n \tag{1}$$

where f = frequency

which defines a straight line on a log-log plot, with slope of n. Van Deusen,[6] in 1967, gave evidence to show that for most natural or man-made surfaces, the slope is roughly -2.

Other researchers have subsequently used the -2 approximation in ride dynamics work, using computer simulation of both the terrain and the vehicle. Before substantive work using a computer simulation is undertaken, the assumptions made when implementing that simulation must be shown not to significantly affect the results of the research. In past studies, several assumptions have been made regarding the computer-generated terrain profiles:

(1) That the profiles pass the test for stationarity. (Murphy used a program called STANOR to actually test his profiles.)[1]

(2) That, as mentioned above, the slope of the PSD curve is -2.0 for any real terrain.

(3) That the input spacing can be set to any arbitrary value (Murphy used 3.07 inches; Kozin used up to 2 feet to estimate the PSD's of real terrain[5]).

The first of these assumptions seems reasonable; especially if the input is generated from random numbers which are stationary. This will be discussed in more detail later. The second two leave some questions unanswered.

In recent work, as yet unpublished, Murphy has measured over sixty actual ground roughness profiles. He has found that the slopes of their PSD curves vary from -0.6 to -2.3. Does the assumption of a -2.0 slope, then, cause significant errors in ride analysis? To answer this question is the first objective of this research. This objective will be attacked by testing the vehicle ride simulation for sensitivity to changes in PSD slope, keeping other statistical parameters constant.

The second assumption concerns a more practical matter. An investigator involved in taking field data for terrain roughness analysis has to trade off time and money considerations against accuracy. Taking data points using transit and rod is time consuming -- thus expensive. Taking data closer together increases cost; while choosing too great a measurement interval decreases accuracy. Theoretically, some point exists where further reduction of interval wastes money. Likewise, at some point a further increase causes a serious reduction in accuracy. The second objective of this report is to obtain some quantitative grasp of the accuracy of vehicle model performance versus distance between data points. This will be accomplished by creating a single terrain profile with a small interval and exercising a computer vehicle simulation over this

terrain at different measurement intervals.

DISCUSSION AND COMPUTER CALCULATIONS

NOIPSD

The first requirement of this study was the generation of terrain profiles, created with random amplitudes. In addition, some control of statistical properties was necessary. A computer program (NOISE1) written for this purpose was obtained from Mr. N. S. Murphy Jr. of the Waterways Experiment Station. Another program, PSD, was designed to accomplish the PSD estimation. To reduce the execution time, NOISE1 and PSD were combined to form the program NOIPSD. Unfortunately, both were written for a GE400 machine, while the computer available was a PDP-10. This necessitated several adjustments in the program, in addition to translation into a different Fortran language. A description of the program and the changes made in its basic structure follows. A program lising is in Appendix A.

First, a random number generator was required. Since these generators (sub-programs) are generally machine-specific, some alteration of the program was necessary. An internal PDP-10 program, called RAN(Z), was used to obtain twelve uniform random numbers between zero and one for each profile point. It was desired that each terrain profile be different. RAN(Z), however, has a fixed starting number for each program execution. This is analagous to having a fixed table of random numbers from which to choose. Obviously, then, each execution would result in the same profile shape. It was decided to use a scheme in which a "skipping number" was used to cause the

program to produce different profiles while maintaining their random nature. Twelve normal random numbers between zero and one are used to obtain random numbers with zero mean by the uniform deviates method:

$$V_j = \sigma_n \left(\sum_1^{12} z_i - 6.0\right) \quad (j = 1,2,3,\ldots,1200)$$

where V_j = a random normal number with zero mean

z_i = a uniform random number from RAN (Z)

σ_n = desired standard deviation of V

Before each twelve numbers used from the "table" of random numbers, NG (the "skipping number") points were skipped. NG should be kept relatively low to reduce computer time. Prime numbers were assigned to NG each time NOIPSD was run to insure profile difference.

After all the V_j's are determined, their mean is computed. It should be zero and usually is very small. To make sure, each V_j is then shifted by the computed mean.

This procedure gives white Gaussian noise, which has essentially a level power spectral density curve. The above spectrum is shaped using a digital simulation of an analog low-pass filter:

$$Y_i = V_i + Y_{i-1}\, e^{-\alpha\tau}$$

where Y_i = the resulting profile

α = the spacial cutoff frequency

τ = the interval between points (constant)

The product $\alpha\tau$ is used to adjust the frequency content of the profile. For most of this study, τ was kept at a constant 4.0 inches. The value chosen for α, then, determined the ultimate power spectral density slope. A table of α values was determined by trial and error and is given below:

Table 1

(Valid for $\tau = 4.0$)

Desired Slope	Use NG of	Use α of
-0.60	2	.2575
-1.20	19	.0888
-1.85	13	.0129
-2.00	7	.0052
-2.15	11	.00138
-2.30	3	.00016

Next, the desired RMS level (DRMS)* is achieved by computing the actual RMS (ARMS) and adjusting each point by the factor DRMS/ARMS. This results in the desired results, and completes the profile generation portion of the program.

Before continuing on in the program description, the mathematical definitions of the autocovariance function and power spectral density estimation need to be presented.

*Names in parentheses indicate program variables. See Appendix A.

Assume $X(t)$ is a continuous, infinite waveform. The <u>covariances</u> (C_{ij}) are defined by:[7]

$$C_{ij} = \text{cov}\left\{X(t_i),X(t_j)\right\}$$

$$= \underset{X}{\text{ave}}\left\{[X(t_i)-\overline{X}(t_i)]*[X(t_j)-\overline{X}(t_j)]\right\}$$

where $\overline{X}$ indicates the mean.

Since the terrain waveforms are created with zero mean, $\overline{X}(t_i) = \overline{X}(t_j) = \overline{X}(t_n) = 0.0$. It follows that:

$$C_{ij} = \underset{X}{\text{ave}}\left\{X(t_i)*X(t_j)\right\}$$

Furthermore, assuming stationarity, the covariances will depend on time separation only.

$$C_{ij} = C(t_i-t_j) = C(\tau)$$

where τ is generally called lag. From this, and assuming ergodic properties, the covariance at lag τ can be written:

$$C(\tau) = \underset{t}{\text{ave}}\left\{X(t)*X(t+\tau)\right\}$$

or, in functional notation:

$$C(\tau) = \lim_{T\to\infty}\frac{1}{T}\int_{-T/2}^{T/2} X(t)*X(t+\tau)dt \qquad (2)$$

which is generally called the <u>autocovariance</u> function. Equation (2) may, in the ideal case, be reduced to:[7]

$$C(\tau) = \int_{-\infty}^{\infty} P(f)*e^{i2\pi f\tau}df \qquad (3)$$

where

$$P(f) = \lim_{T\to\infty} \frac{1}{T} \left| \int_{-T/2}^{T/2} X(t)*e^{-i2\pi ft}dt \right|^2 \qquad (4)$$

which defines P(f), the <u>power spectral density</u> of the X(t) waveform. Note that t, the independent variable, may be either time or distance.

If Equation (3) is inverted, the power spectral density (PSD) is seen to be the Fourier transform of the autocovariance function:

$$P(f) = \int_{-\infty}^{\infty} C(\tau)*e^{-i2\pi f\tau}d\tau$$

Since $C(\tau)$ is an even function (symmetric about the zero axis), the equation may be simply written:

$$P(f) = 2\int_{0}^{\infty} C(\tau)*\cos(2\pi f\tau)d\tau$$

In the particular case considered herein, the data to be analyzed is discrete and equally spaced. One very important problem to be considered in these cases is aliasing. If the data is taken from a continuous waveform at equally spaced intervals, as would be the case in recording a real terrain, some frequency estimates are distorted. The basic problem is illustrated in Figure 2 which shows

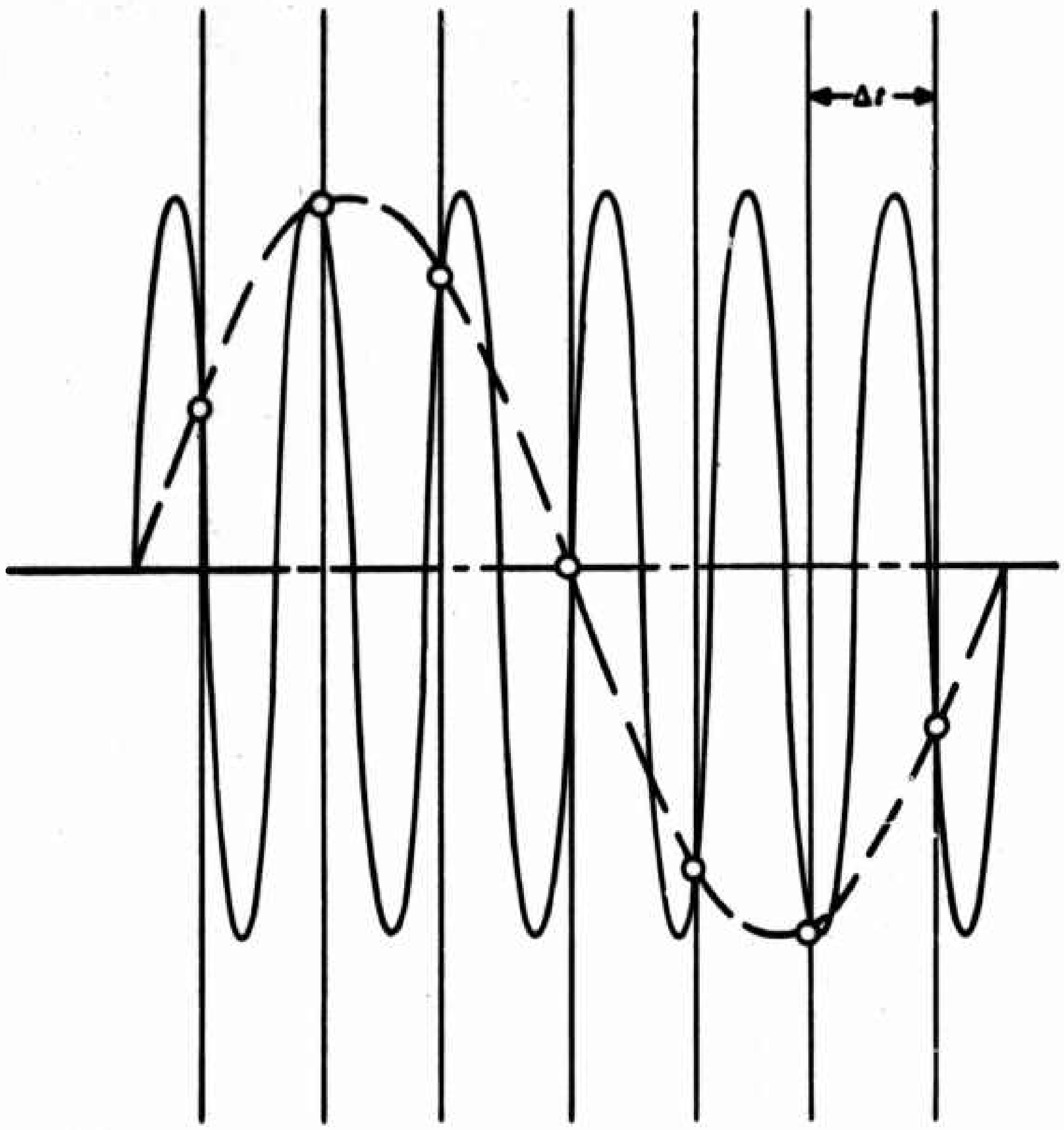


Figure 2. Example of Aliasing

how, in equi-spaced sampling of a sine wave, an imaginary longer wavelength appears to be present. In the present case, the discrete terrain data are not measured from a continuous source, but created in its discrete form. The highest frequency possible is $1/(2\Delta t)$ which is used as the high cut-off frequency. Aliasing, in this special case, is of no concern.

For discrete data, Blackman and Tukey suggest a three-step method for power spectral density estimation:[7]

(1) If $X(1),X(2),X(3),\ldots,X(N)$ is the discrete series with interval $\Delta\ell$ between values, the autocovariance is computed with lag $\tau = h*\Delta\ell$ as follows:

$$C(\tau) = \frac{1}{N-hr} \sum_{q=0}^{N-hr} \left(X(q)*X(q+hr) \right) \qquad (5)$$

where N = number of points

$r = 0,1,2,\ldots,m$

$m \leq N/h$ = maximum lag

$h > 0$ (integer)

(2) The raw spectral estimate is computed:

$$V(r) = \Delta\tau \left[C_0 + 2 \sum_{q=1}^{m-1} \left(C_q*\cos(qr\pi/m) \right) + C_m*\cos(r\pi) \right] \qquad (6)$$

where $f(r) = r/(2m\Delta\tau)$

$r = 0,1,2,\ldots,m$

$C_i = C(\tau_i)$

(3) Due to the finite number of data to be analyzed and the discrete number of lags, the spectral estimate must be smoothed. Hamming is used in this analysis:

$$U(r) = A_{io}V_r + \sum_{j=1}^{m} A_{ij}\left[V_{r+j} + V_{r-j}\right] \qquad (7)$$

where, for hamming: (i=3)

$A_{30} = 0.54$

$A_{31} = 0.23$

$A_{3j} = 0.00$ for $j = 2,3,4,\ldots$

Returning to the computer program, NOIPSD, the first step in the PSD estimation process is to compute the maximum number of lags (LLAG). From Blackman and Tukey the equation is:[7]

$$K = \frac{2\left(\frac{N}{\tau} - \frac{LLAG}{3}\right)}{LLAG}$$

where K = the equivalent number of degrees of freedom of a chi-square distribution

N = number of points in profile

LLAG = number of lags (max lags)

τ = profile spacing

Solving the above equation for the number of lags:

$$LLAG = \frac{6N}{\tau(3K+2)}$$

Obviously, a balance exists among the number of points, N, the number of lags, and K. The choice of K determines the degree of accuracy of the estimate, as seen in Figure 3. It shows the distribution of PSD estimates, for instance, as fixed multiples of their average values. As an example, consider a PSD estimate which has an average value of 10 in^3/cycle. For K = 10, individual estimates will, in the long run, fall below .49 times its average value (4.9 in^3/cycle). In brief, 80 percent of all values would fall within the interval 4.9 to 16.0. As can be seen, higher values of K give higher accuracy. But for a fixed N, the number of lags decreases with K, reducing the amount of information in the PSD curve. Thus with a very high K, one point on the PSD curve would result. It would be very accurate, but not of much use. For N = 1200 and τ = 4.0, a value of K = 20 gives 30 lags. This value of K gives reasonable accuracy of estimates while giving a reasonable number of lags. Since the program must allow for changes in N and τ , the equation was written:

$$m = LLAG = 6N/(62*\tau)$$

which is used in the program.

The autocovariance computation is done by using h=1 in equation (5). A check is made on the RMS level at this point by taking the square root of the first (r=0) autocovariance, which should equal the desired RMS.

Next, the raw spectral estimates (PX) are computed using equation (4), and smoothed by hamming to give the smoothed power

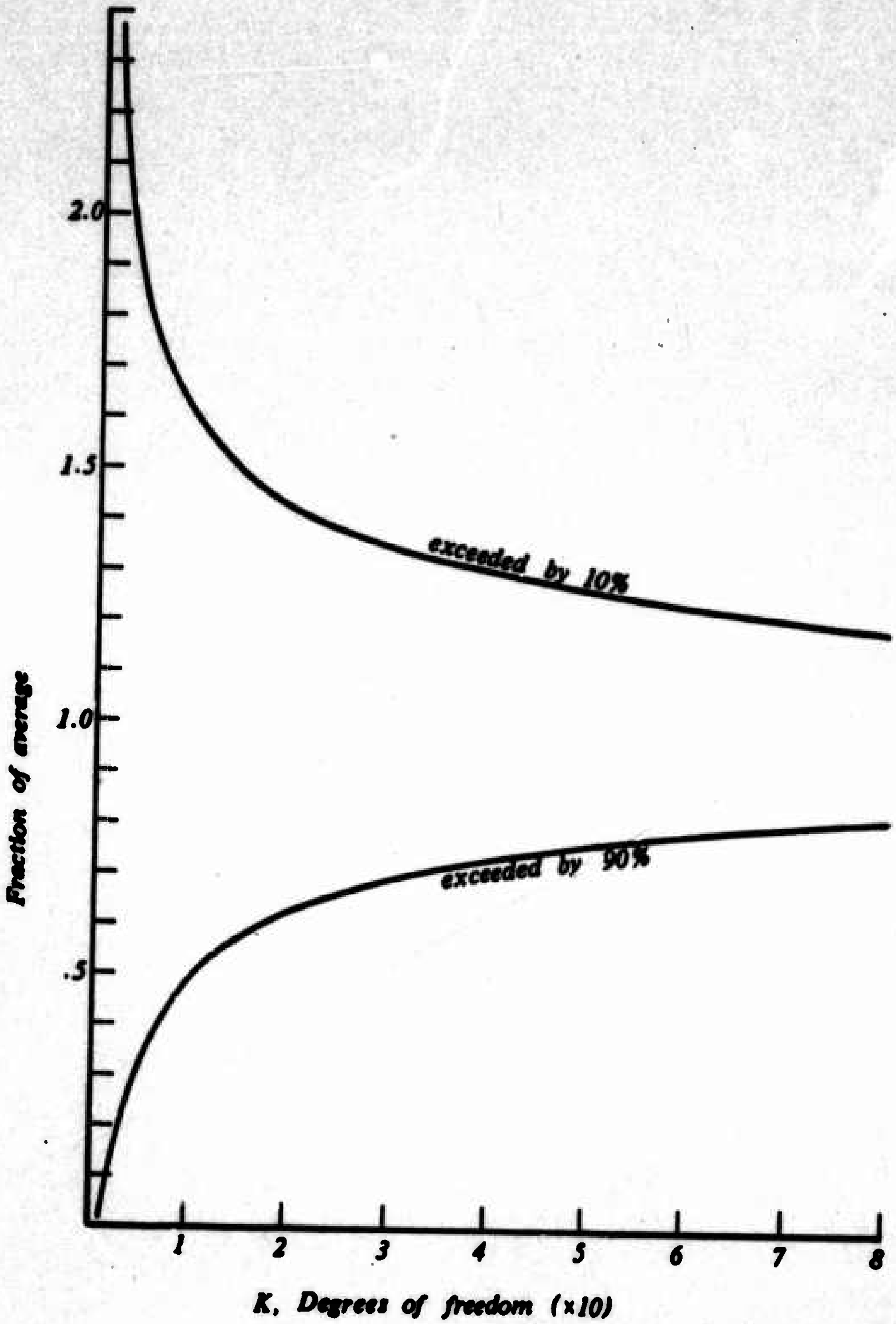


Figure 3. Relative Accuracy vs.
Equivalent Degrees of Freedom

spectral density estimates (SPX):

$$SPX(1) = .54PX(1) + .46PX(2)$$

$$SPX(j) = .23PX(j-1) + .54PX(j) + .23PX(j+1) \quad (j=2,3,4,\ldots,m-1)$$

$$SPX(m) = .54PX(m) + .46PX(m-1)$$

Finally, after discarding PSD points outside the low and high cut-off frequencies, the smoothed PSD estimates are fitted to a straight line on a log-log plot, using a least square fit routine. The result is an equation:

$$PSD = Cf^n$$

where C = the intercept of the line at f=1.0

f = frequency

n = slope of the line

This completes the discussion of NOIPSD.

VEH

The next major requirement of this work was a vehicle simulation. Again this was, in its original form, obtained from Mr. N. S. Murphy Jr., WES. The program was also translated to PDP-10 FORTRAN IV for use herein. Several subprograms were changed or added for this study, including DATA, GAMSUB, WHEELS, PRINT, and FILIN. Basically, this computer program is a 2-dimensional, 5-degree-of-freedom simulation.

The following assumptions were made in deriving the equations of motion:

(1) The vehicle sprung mass is rigid.

(2) The only forces acting on the sprung mass are suspension forces.

(3) There is no surge acceleration or motion in sway, yaw, or roll degrees of freedom.

(4) Pitch is small enough to allow a small-angle approximation.

Assumption 3 was made reluctantly, but was necessitated by time and cost considerations. The others are reasonable considering the purpose of the simulation.

Figure 4 shows a schematic of the general wheeled vehicle model. Three axles are shown and that is the maximum number which can be simulated with this model. A two-axle vehicle is modeled by setting ℓ_3, ℓ_4, and m_3 equal to zero. Thus, the one model can simulate almost any conventional vehicle. In this study, one 2-axle and one 3-axle vehicle are used. They are the M151 Jeep and the M35 2½T Truck, respectively.

The equations of motion were written using Newton's Law on each of the free-body masses shown in Figure 5.

Body:

Sum of vertical forces:

$$m_o\ddot{z} = F_1 + F_2' - m_o g \qquad (8)$$

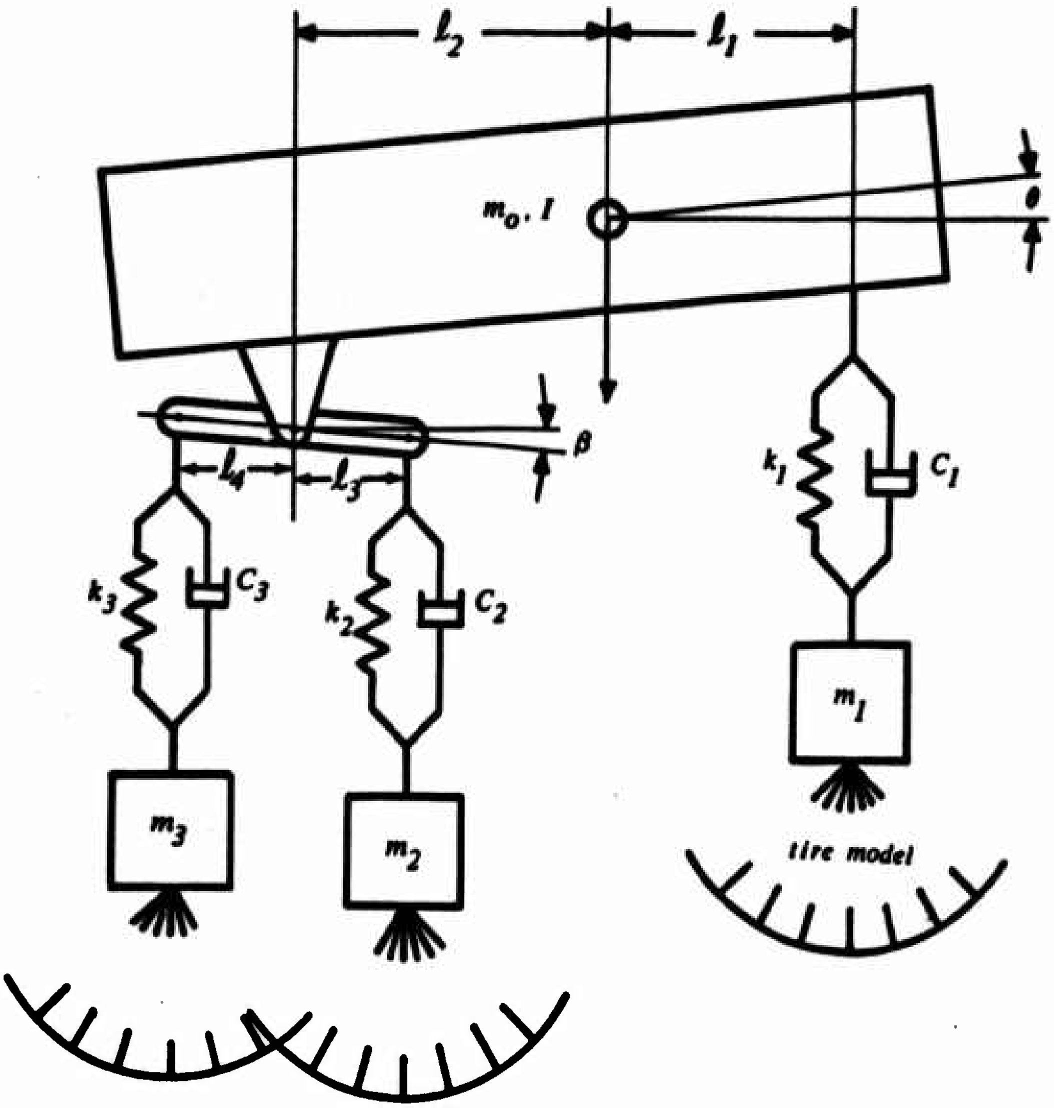


Figure 4. Schematic of General Wheeled Vehicle Model

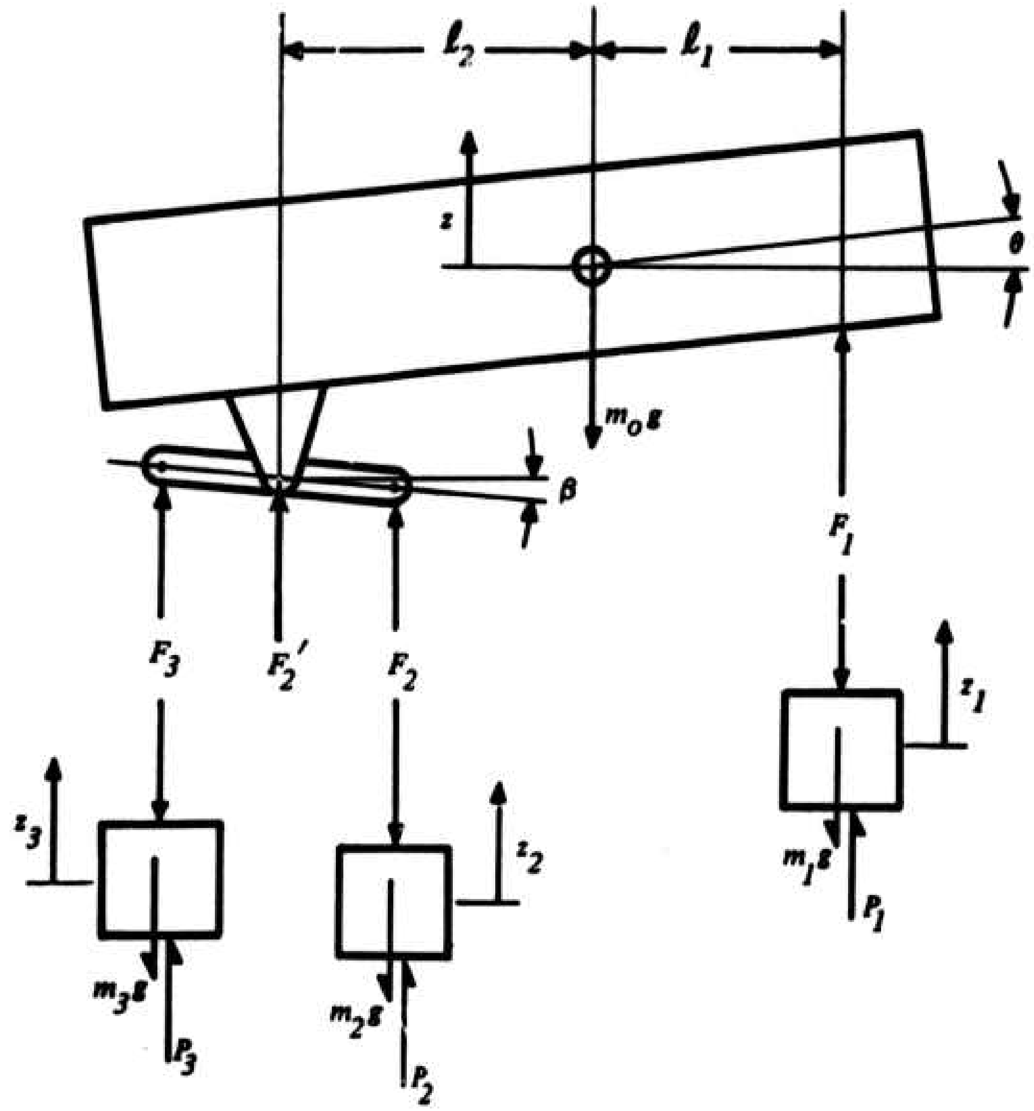


Figure 5. Free Bodies of General Wheeled Vehicle Model

Sum of moments about pitch axis:

$$I\ddot{\theta} = F_1\ell_1 \cos\theta - F_2'\ell_2 \cos\theta \tag{9}$$

Unsprung masses:

Sum of vertical forces:

$$m_i\ddot{z}_i = -F_i - m_i g + P_i \quad (i=1,2,3) \tag{10}$$

The angle β is defined in the rear suspension geometry as in Figure 5. The forces F_1, F_2', and P_i are defined:

$$F_1 = k_1\Delta_1 + C_1\dot{\Delta}_2$$

$$\Delta_1 = z_1 - z - \ell_1 \sin\theta$$

$$\dot{\Delta}_1 = z_1 - z - \ell_1 \dot{\theta} \cos\theta$$

$$F_2' = F_2 + F_3$$

$$F_2 = k_2\Delta_2 + C_2\dot{\Delta}_2$$

$$\Delta_2 = z_2 - z + \ell_2 \sin\theta - \ell_3 \sin\beta$$

$$\dot{\Delta}_2 = \dot{z}_2 - \dot{z} + \ell_2\dot{\theta} \cos\theta - \ell_3 \dot{\theta} \cos\beta$$

$$F_3 = k_3\Delta_3 + C_3\dot{\Delta}_3$$

$$\Delta_3 = z_3 - z + \ell_2 \sin\theta + \ell_4 \sin\beta$$

$$\dot{\Delta}_3 = \dot{z}_3 - \dot{z} + \ell_2\dot{\theta} \cos\theta + \ell_4 \dot{\beta} \cos\beta$$

P_i = forces from terrain profile as transmitted by the tire model.

The program named VEH is designed to solve equations 8, 9, and 10. Basically, it performs its function using a fourth order Runge-Kutta-Gill algorithm. A program listing is in Appendix B.

Before the calculations are begun, program options and variable parameters are fed into the program, normally by teletype on a remote hookup. The program options determine which parts of the program will be executed. The variable parameters include the input profile name, tire deflections, vehicle velocity in miles per hour and the teletype printout time interval. These parameters will be explained more fully in the text that follows.

First, VEH calls subroutine FILIN. On this initial call, FILIN requests an input of desired $\Delta\ell$ (DELTAL)*, which may be any integer multiple (≥ 1) of the input profile spacing. FILIN then reads the actual input profile spacing (SPACING), the profile identification line (FID) and the first ten profile points (FYIN) from the input file. A parameter, MM, is then calculated by the following formula:

$$MM = \frac{DELTAL}{SPACING}$$

which defines the number of profile points (FYIN) to be skipped between each two points returned to the main program. Thus, if the spacing in the input profile is 1.0 inches and the desired spacing is 4.0 inches, MM = 4. Every fourth point will be returned to the main program. This feature was added to allow investigation in this study

*The words in parentheses are the parameter names in the program listing. See Appendix B.

of the effect of input profile spacing on simulation performance.

After the vehicle name and the program options are read, the program calls subroutine DATA. DATA sets the vehicle parameters to the proper values. It must contain every vehicle parameter necessary to simulate the desired vehicle. Some of the data statements contain values of interest only for tracked vehicles. The tracked vehicle subroutines are not included, but the data was left in case later investigators wish to use the program. The tracked vehicle subroutines are available if desired. For wheeled vehicle simulation, a list of necessary vehicle parameters for inclusion in subroutine DATA is given in Appendix C.

After most of the parameters are set, but before returning to the main program, DATA calls subroutine GAMSUB, which computes the parameters used in the radial-spring tire model. The tire model is constructed under the assumption that a tire can be simulated by a series of radial springs, as in Figure 6. In the original program, the calculations done in GAMSUB were made by hand and inserted through subroutine DATA. In this study it was desired to change the input profile spacing from one execution to the next. Thus, GAMSUB takes into account the input profile spacing (DELTAL) and "creates" radial springs such that the distance between the projections of their outer ends on a horizontal surface will be exactly equal to the profile spacing. Thus, $D_1, D_2, \ldots, D_{kk}$ are multiples of DELTAL. The maximum angle allowed from the vertical is roughly 53 degrees. One spring is always placed vertical, the others symmetrical to front and rear.

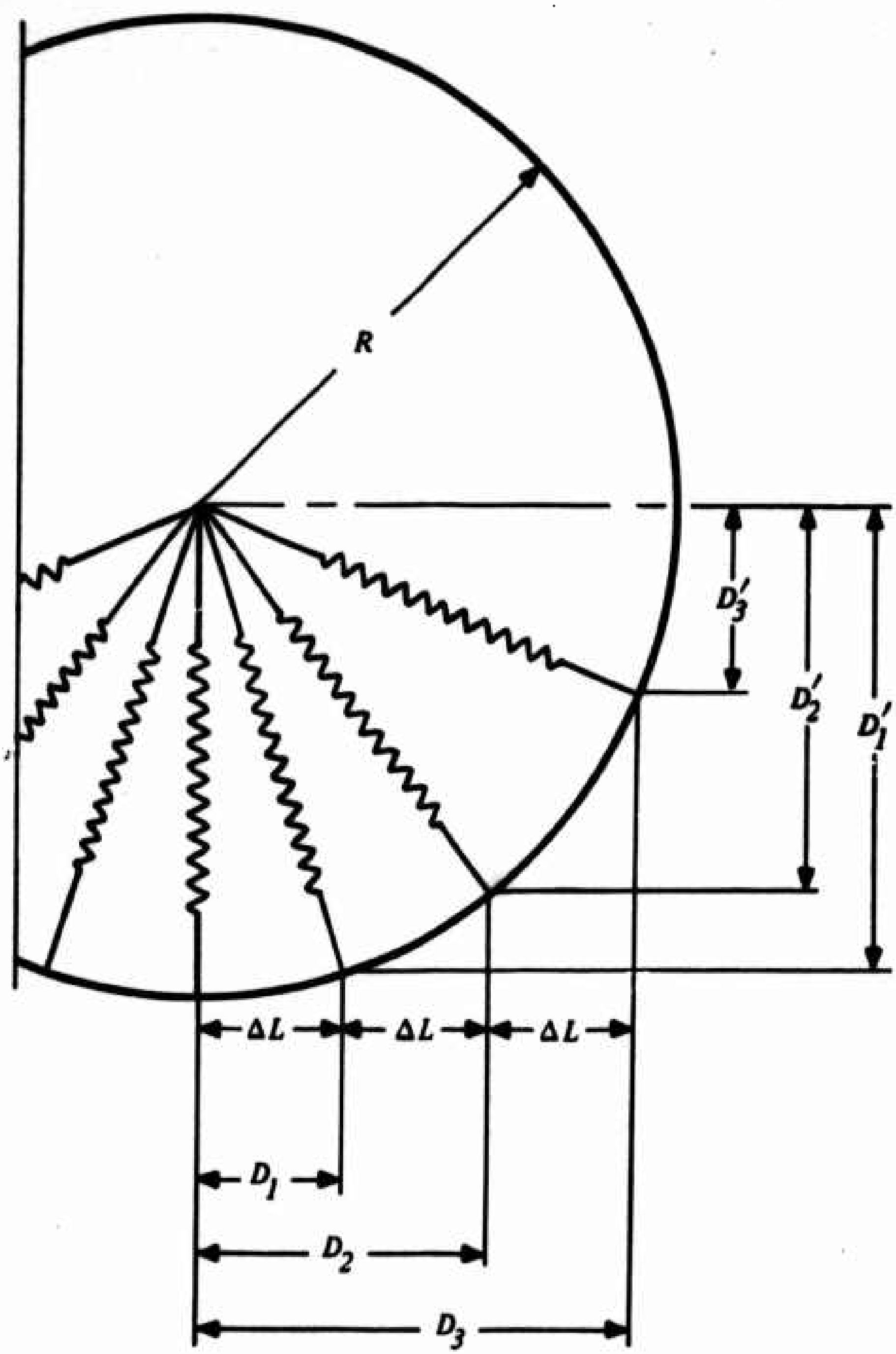


Figure 6. Tire Model
Spring Locations

Basically, γ (GAMMA) is the vertical component of the radial spring force-deflection function. Each radial spring is assumed to have the same spring rate, which is determined separately for each tire from the tire force-deflection curve. The curves for M151 tires, for instance, are shown in Figure 7. The curves are very close to being straight lines, so a linear approximation is used. The load on each tire is computed (note that for 3-axle vehicles the rear tires are assumed to be duals) and typed out, in order from front to rear, on the teletype. After each load, a tire deflection is entered by the operator. Care must be taken to enter the deflection at the proper inflation pressure. The M151 Jeep, for instance, should have 18 psi in the front tires and 22 psi in the rear for cross-country operation.[8] GAMSUB then computes the radial spring force for each tire. SPKF, SPKR1, AND SPKR2 are the program variables for these spring forces front to rear. They are computed in the following manner:

Suppose the dead load on the tire causes a tire deflection (Y) as in Figure 8. The radial spring constant (SPK) is computed from the dead load (WEIGHT) and spring deflections (DELTA) as follows:

$$\text{WEIGHT} = \sum_{i=1}^{\text{NSEGS}} \text{SPK}*\cos\,\varphi_i*\text{DELTA}_i \qquad (11)$$

where SPK*cos θ_i = the vertical component of SPK

NSEGS = 2*KK+1 , the number of radial springs in the tire model.

$$\text{SPK} = \frac{\text{WEIGHT}}{\sum_{i=1}^{\text{NSEGS}} \cos\,\varphi_i*\text{DELTA}_i} \qquad (12)$$

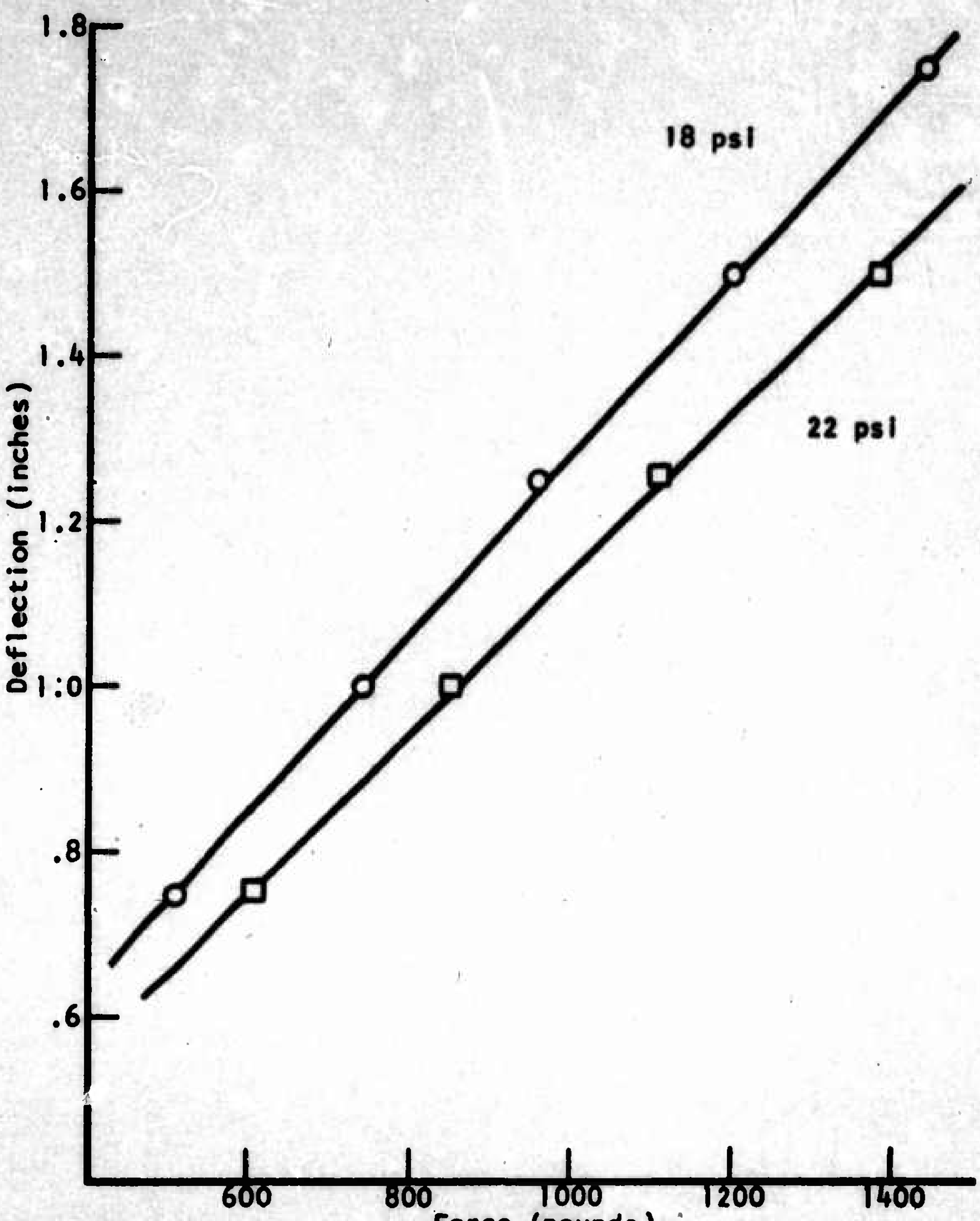


Figure 7. Tire Force-Deflection Curves for M151

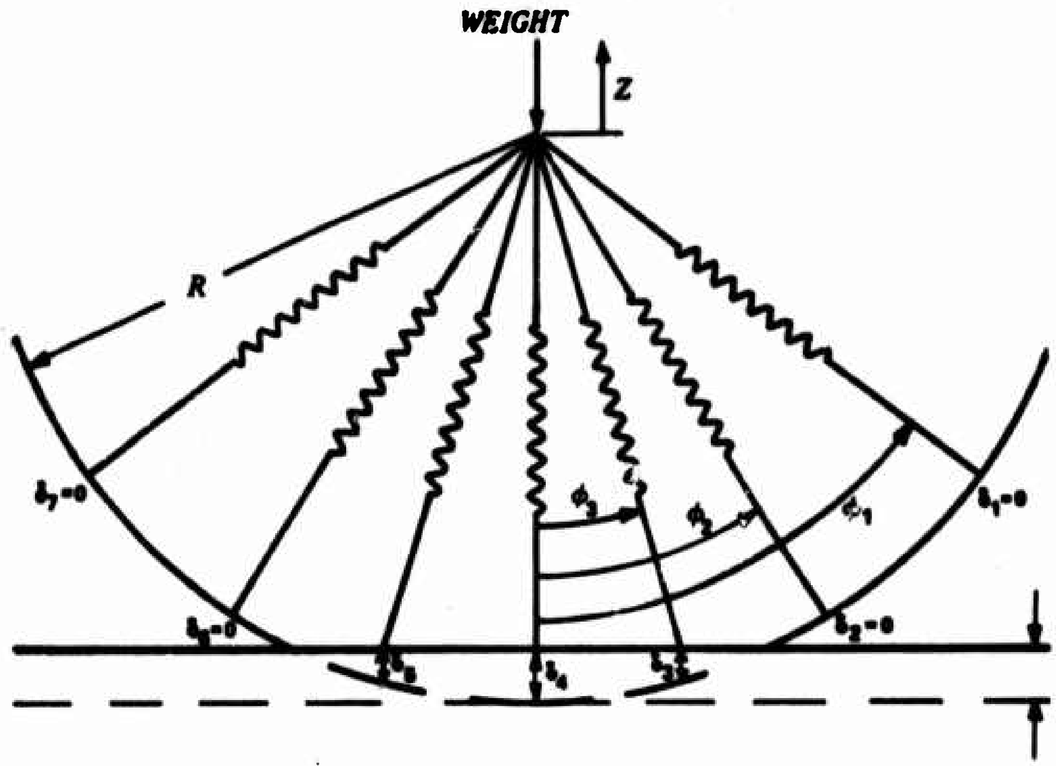


Figure 8. Tire Deflection with Dead Load

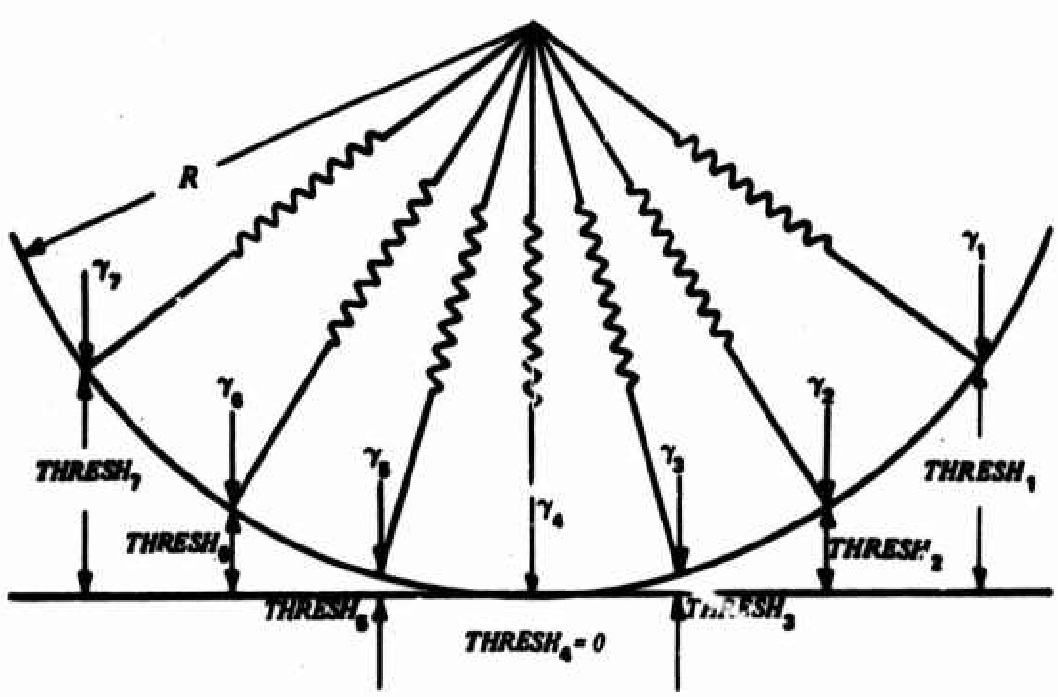


Figure 9. Final Elements of Tire Model

since SPK is assumed constant. In addition, since $\cos\theta_i$ can be written (see Figure 6):

$$\cos\theta_i = \frac{D_i'}{R}$$

equation (9) becomes:

$$SPK = \frac{WEIGHT}{\sum_{i=1}^{NSEGS} \frac{D_i'}{R} * DELTA_i} \tag{13}$$

where $DELTA_i = Y - THRESH_i - z$

(if $DELTA_i$ is < 0.0, $DELTA_i = 0.$)

$THRESH_i = R - D_i'$ (see Figure 9)

R = the undeflected tire radius

Note that in this case, Y causes deflections only in three of the radial springs. Thus, $DELTA_1 = DELTA_2 = DELTA_6 = DELTA_7 = 0.0$ and equation (11) becomes:

$$SPK = \frac{WEIGHT}{\frac{D_3'}{R} * DELTA_3 + \frac{D_4'}{R} * DELTA_4 + \frac{D_5'}{R} * DELTA_5}$$

The radial spring rates for different tires on the same vehicle may vary, either because of different inflation pressures or different load distribution.

As the final step, GAMSUB computes the values for GAMMA by the equation:

$$GAMMA_{ij} = SPK_j * \cos \theta_i$$

$i=1,2,3,\ldots,NSEGS$

$j=1,2,3$ (axle number)

For computer notation, the GAMMA's are numbered from front to rear of the vehicle as shown in Figure 10.

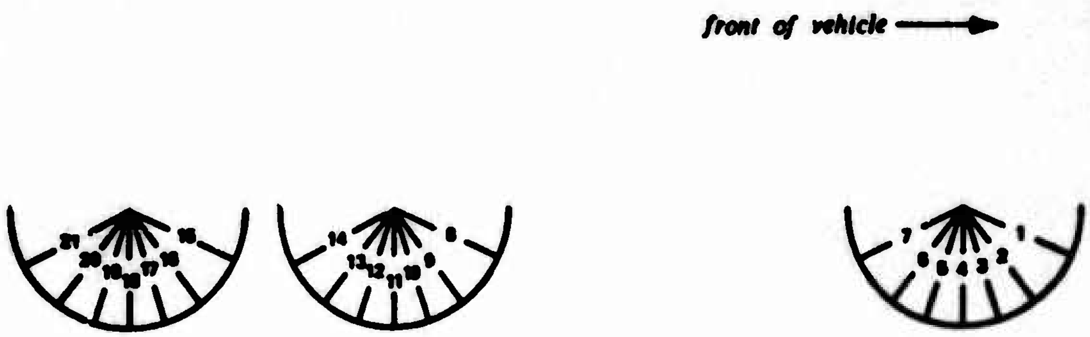


Figure 10: Numbering Scheme for Tire Segments

Upon return to the main program, the number of steps to be used in the Runge-Kutta-Gill (RKG) algorithm is computed, based of vehicle velocity. The step size is desired to be roughly .001, which is adjusted slightly to insure an exact number of steps between input terrain profile points. This completes the preliminary calculations.

The program enters an integration loop at this point, starting effectively with the calling for the second time of subroutine FILIN. Each time FILIN is called from the integration loop, it simply returns one profile point (YIN) to the main program.

If a detailed output file is desired, the program calls FILWRT, which writes time (T), YIN, and absorbed power (ABSPWR) for each step. It also writes displacement, velocity, acceleration, and RMS

The force exerted through the suspension system to the body must now be computed for each axle.

WHEELS allows considerable flexibility in modeling spring-deflection-vs.-force curves and in modeling damping-force-vs.-deflection-velocity curves. Both curves are specified by data statements in subroutine DATA.

The spring force-deflection function of a typical suspension spring can be approximated by five linear segments, as shown in Figure 11.

The deflection axis can be divided into five regions:

Region 1: $-\infty$ to x_1

Region 2: x_1 to x_2

Region 3: x_2 to x_3

Region 4: x_3 to x_4

Region 5: x_4 to $+\infty$

The x_i are the region limits ($SLIMIT_i$).

In each region, the force-deflection function is approximated by a linear equation of the form:

$$FORCK_i = m * SPDEF_i + C$$

where FORCK = resultant spring force

m = slope of the line (SSLOPE)

SPDEF = deflection of the spring

C = intercept of the line at SPDEF = 0.0 (SINT)

i = axle number

acceleration of the center of gravity, the pitch, and each axle. If the teletype printout interval is exceeded, subroutine PRINT is called, which causes the same information to be typed out. If desired, PRINT will cause only time and absorbed power to be printed out. PRINT also asks the operator if he desires to stop execution. If the answer is yes, the main program transfers out of the integration loop. If tha answer is not yes, the main program then calls subroutine SHIFT.

SHIFT, as the name implies, causes each profile point to be shifted by DELTAL inches to the rear of the vehicle. The main program then sets the first profile point to equal YIN and continues.

Next, the differential equations are solved. For each RKG step between profile points, the suspension forces are calculated by a subroutine called WHEELS.

The first thing done by WHEELS is the computation of the forces on each axle (FORCW). It uses the same method as before to determine radial tire-model spring deflections, DELTA:

$$DELTA_i = Y_k - THRESH_i - Z_j$$

where j = axle number

no negative values of DELTA are allowed

Y_k = elevation of the profile point under the k-th spring

The axle forces (FORCW) are:

$$FORCW_j = \sum_{i=1}^{NSEGS} DELTA_i * GAMMA_i \qquad (j=1,2,3)$$

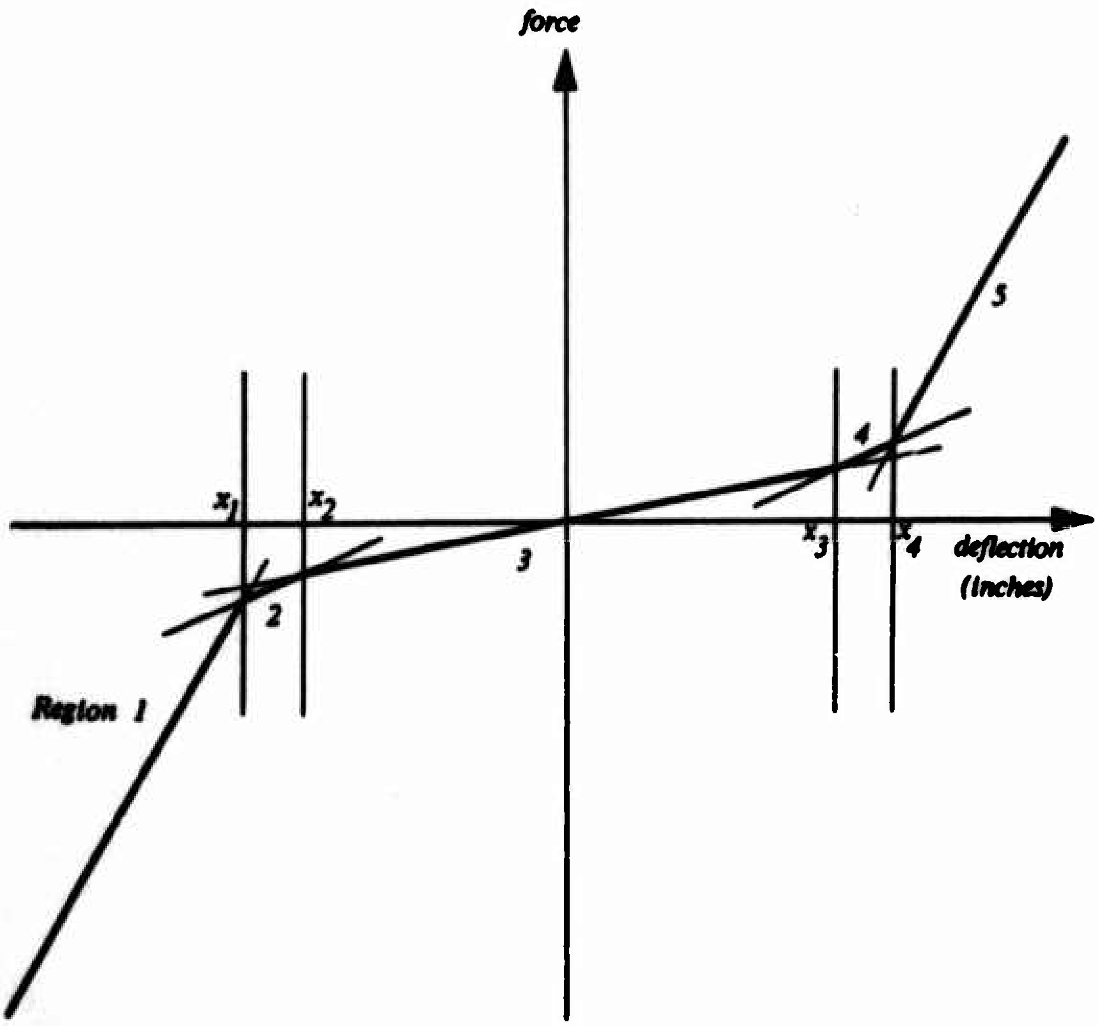


Figure 11. Segmented-Linear Spring Force Approximation

The simulation of the suspension damping function is essentially the same. Details are given in Appendix C.

Thus, after computing the suspension spring deflection (SPDEF) and deflection velocity (DSPDF), subroutine WHEELS compares their values with the region limits, decides which equations to use, and computes the suspension forces (FORCK and DAMP).

Execution then returns to the main program, which immediately calls subroutine RUNGE, the RKG integration scheme. Next, if the absorbed power option is requested, subroutines POWER and RUNGE are used to compute it.

The concept of absorbed power in mobility research appeared as an attempt to resolve the confusion in driver vibration limits exhibited by Figure 1. It was theorized by Lee, Pradko and others[3,4] that the driver will adjust the vehicle conditions (e.g., speed) so that he can completely absorb, by flexing and unflexing his muscles, all the power of the vibrations he receives from the vehicle in order to keep his eyes or hands steady to see clearly and operate the vehicle controls. A series of experiments indicated a great deal of merit in the concept and determined that 6 watts was a general level of power that the human can or is willing to absorb during driving. This has been accepted by many investigators in ride dynamics, not so much as an ultimate truth but as the best available now.

In the course of these studies several ways of calculating absorbed power were developed (the first three reported in Lee and Pradko[3]):

A. For infinite averaging time:

$$\text{average absorbed power} = \lim_{T\to\infty} \frac{1}{T}\int_0^T F(t)v(t)dt$$

where: F(t) = input force to the driver support (e.g., seat)

V(t) = input velocity of the driver support

B. For a finite averaging time:

$$\frac{1}{\omega_n^2}\frac{d^2P(t)}{dt^2} + \frac{dP(t)}{dt}\left(\frac{2\delta}{\omega_n}\right) + P(t) = KF(t)\,V(t)$$

P(t) = finite average absorbed power

δ = damping factor

ω_n = lowest frequency to be averaged (rad/sec)

F(t) = input force to the driver support

V(t) = input velocity to the driver support

K = conversion constant

C. In the frequency domain, it can be computed:

$$P = \sum_{i=0}^{N} K(f_i)(RMS_A(f_i))$$

where P = finite average absorbed power

$K(f_i)$ = parameter dependent on frequency i

$RMS_A(f_i)$ = RMS acceleration at frequency i

Tables of $K(f_i)$ are given in Reference 3.

D. Recently, work has been done at WES to create an algorithm to compute absorbed power in the time domain. The result was the following

set of equations, derived from the analog circuits shown in Figure 12. a(t) is the acceleration of the driver support, the other variables may be considered as intermediate quantities needed in the calculation of absorbed power.

$$\dot{x}_{11} = -0.1755\ a(t) - 2.742\ u_2$$

$$\dot{x}_{10} = -1.755\ a(t) - 388.8\ x_{11} - 46.67\ x_{10}$$

$$u_2 = -0.1755\ a(t) - 1.042\ x_{10}$$

$$\dot{x}_9 = -10\ u_2 - 6.249\ u_1$$

$$\dot{x}_8 = -10\ u_2 - 78.59\ x_9 - 55.28\ x_8$$

$$u_1 = u_2 - 3.246\ x_8$$

$$\dot{x}_7 = -100\ u_1 - 47.78\ u_o$$

$$\dot{x}_6 = -10\ u_1 + 71.6\ x_7 - 53.49\ x_6$$

$$u_o = -\ u_1 + 1.318\ x_6$$

$$\dot{x}_5 = -100\ u_o - 59\ x_5$$

$$\dot{x}_2 = -0.01294\ a(t)$$

$$\dot{x}_1 = 0.00873\ x_2\ x_5$$

$$PWR = \frac{100\ x_1}{t}$$

In a study yet to be published, Murphy has found this latter algorithm to be accurate within the frequency range of interest (.50 to 20 cps). His version of subroutine POWER was used in the program.

After the integration scheme is completed for each step, the output scaling is accomplished, RMS accelerations are calculated, and a new profile point is obtained from FILIN to begin the integration loop anew.

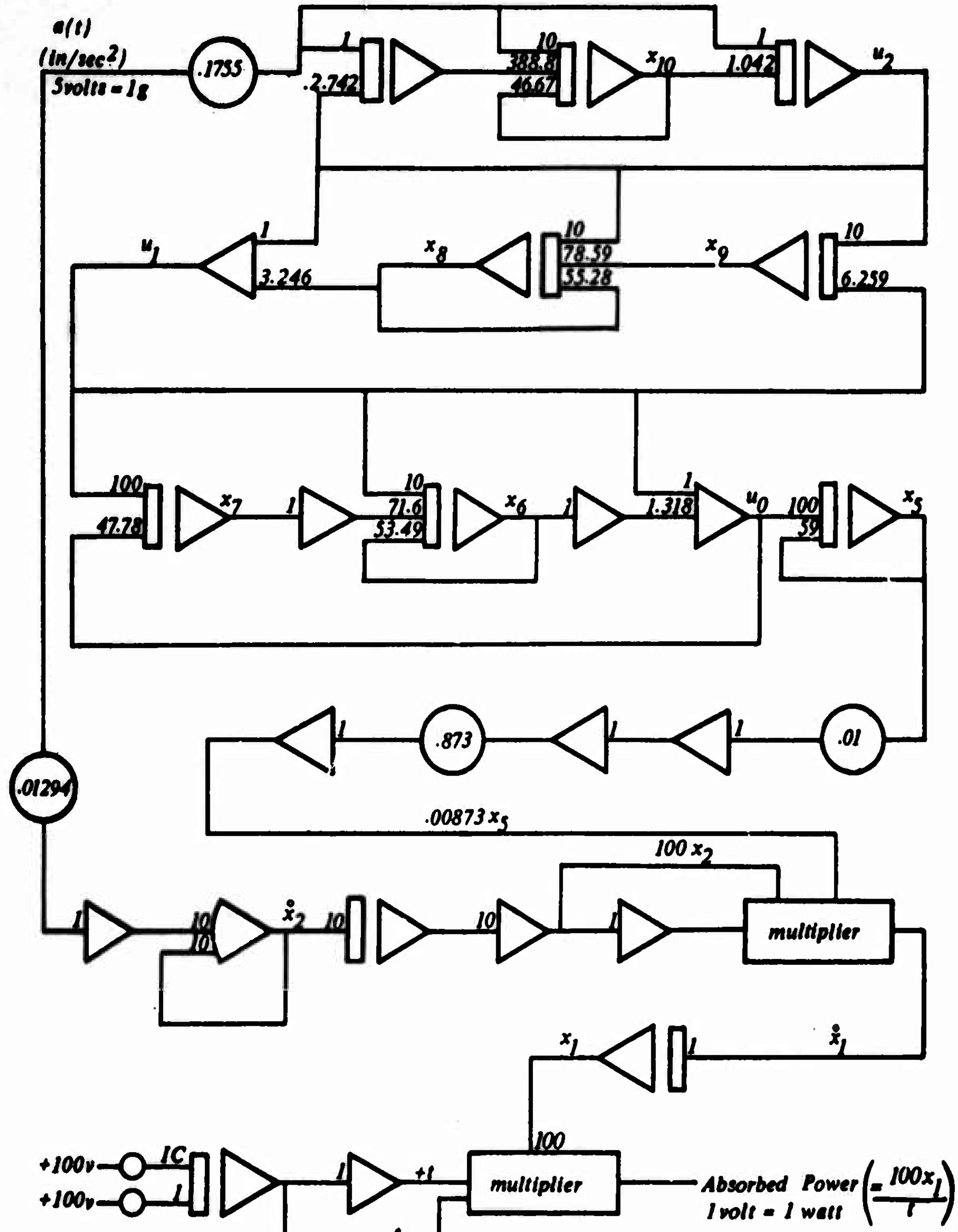


Figure 12. Analog Circuit for Absorbed Power Computation given by Murphy[1]

TEST PROCEDURE AND RESULTS

1. The first objective, as stated previously, was to test the sensitivity of the vehicle simulation to changes in the PSD slope of the input terrain profile.

Six terrain profiles were created, using NOIPSD, with PSD slopes ranging from -0.6 to -2.3. Each profile had 1200 points spaced at 4-inch intervals and an RMS elevation of 4.0 inches. The inputs to NOIPSD for creation of these profiles are repeated in Table 2.

Table 2

(valid for $\tau = 4.0$)

Desired Slope	Use NG of	Use α of
-0.60	2	.2575
-1.20	19	.0888
-1.85	13	.0129
-2.00	7	.0052
-2.15	11	.00138
-2.30	3	.00016

For PSD estimation, a low cut-off frequency (FLOW) of .0002 was used in each case. This caused the zero-frequency estimation to be deleted while all others were used in the curve-fitting routine. The zero-frequency (or infinite wavelength) component of the terrain profile would have no effect on the vehicle simulation. The resultant

PSD data and equations are shown in Figures 13 through 18. Figure 19 shows all six equations superimposed for comparison.

Figures 20 through 22 show the initial 40 feet of the terrain profiles. The vertical scales of the profiles are distorted by a factor of eight, so they appear rougher than they should. The high frequency component can be seen to decrease as the PSD slope becomes larger, which conforms to the expected trend.

Originally each vehicle was to be exercised over each profile, and by trial and error, a "critical speed" determined. This critical speed was defined as that speed at which the vehicle exhibited six watts absorbed power at the driver's seat. The M151 Jeep simulation was exercised first. Each simulation was run until the absorbed power seemed to stabilize (until several consecutive values were very close). This normally took from 4-6 seconds of real time. The following results were thus obtained:

Table 3

PSD Slope	M151 Critical Speed (mph)
-0.60	2
-1.20	2
-1.85	4
-2.00	5
-2.15	10
-2.30	17

This data is plotted in Figure 21. At the end of one of the last runs,

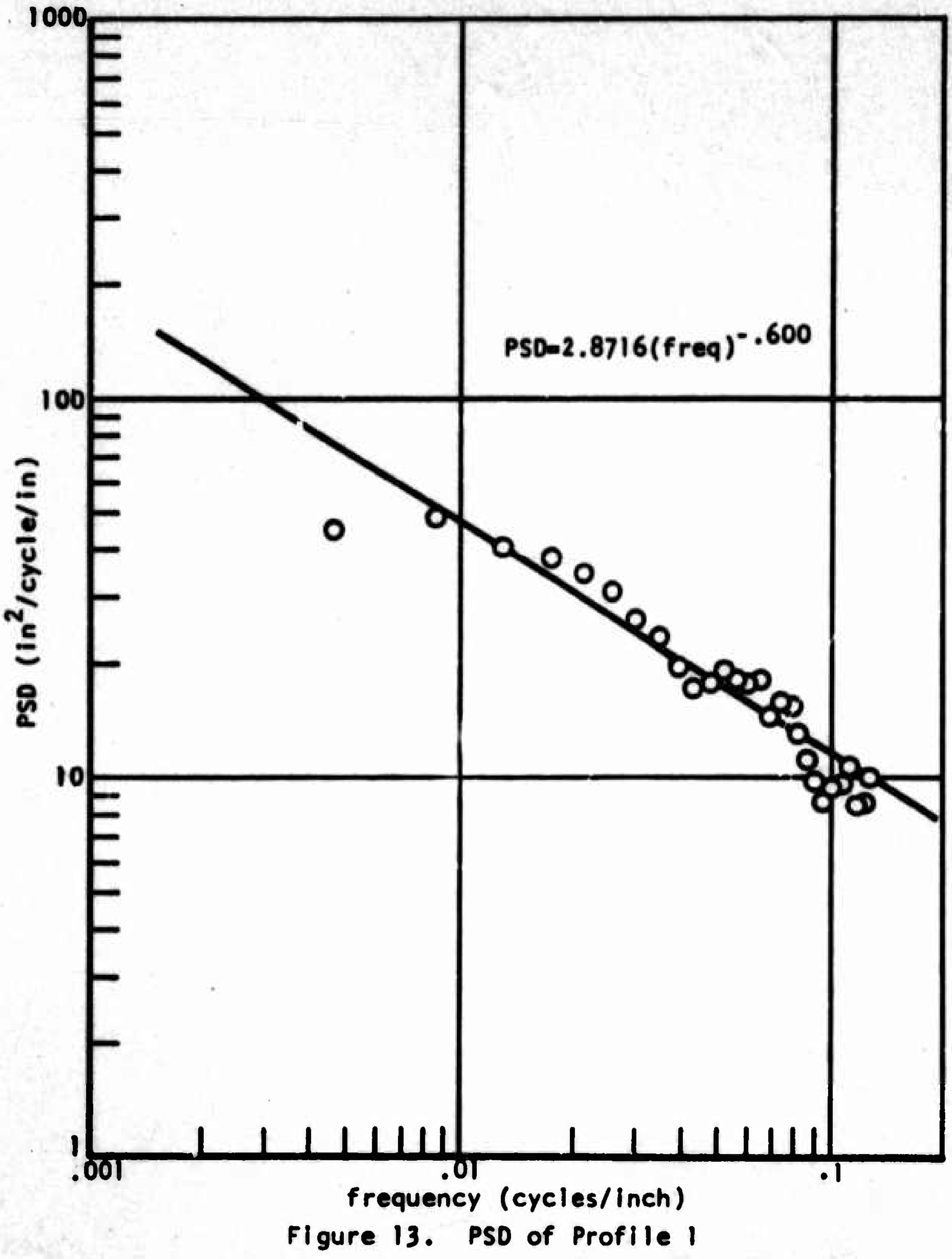


Figure 13. PSD of Profile 1

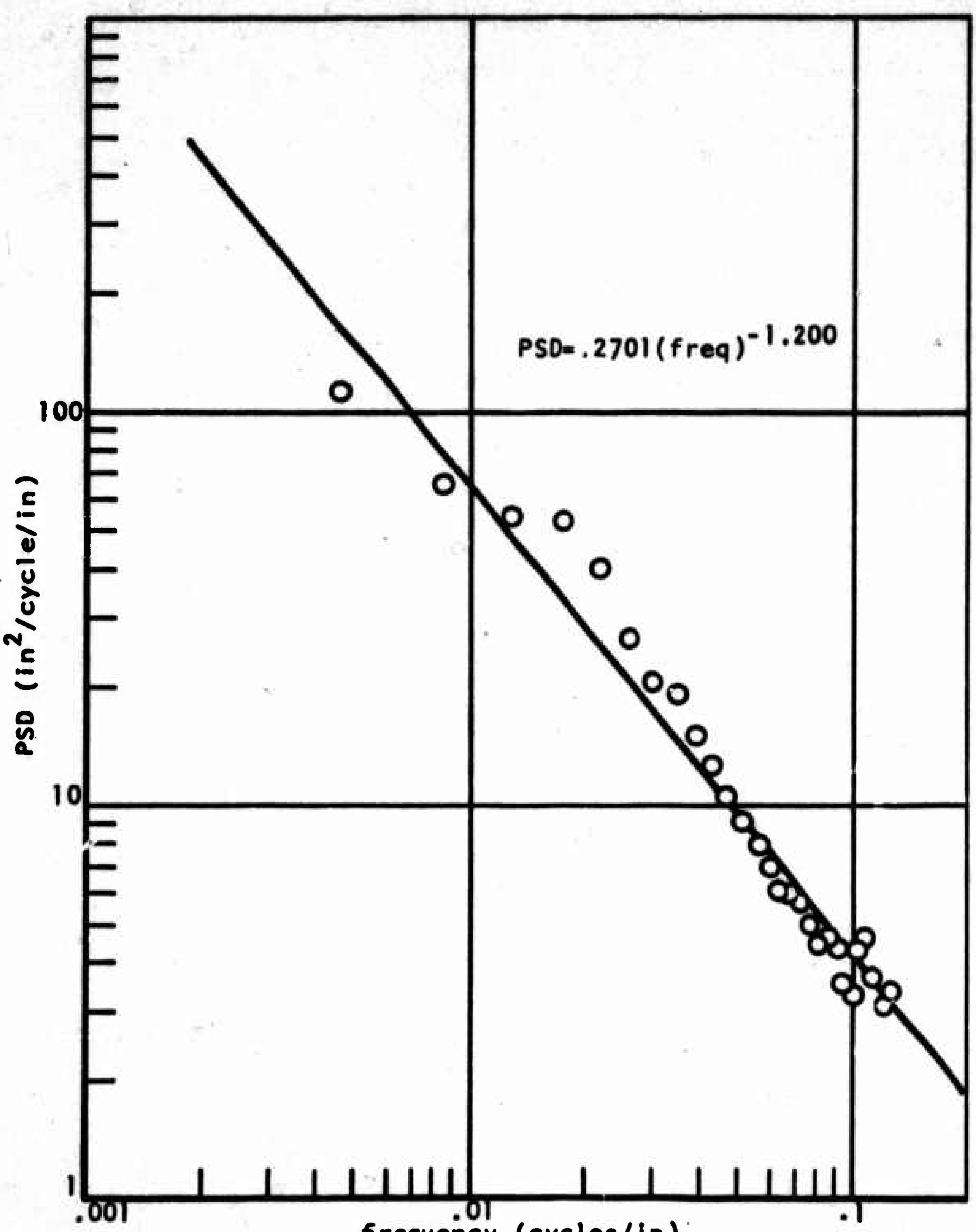


Figure 14. PSD of Profile 2

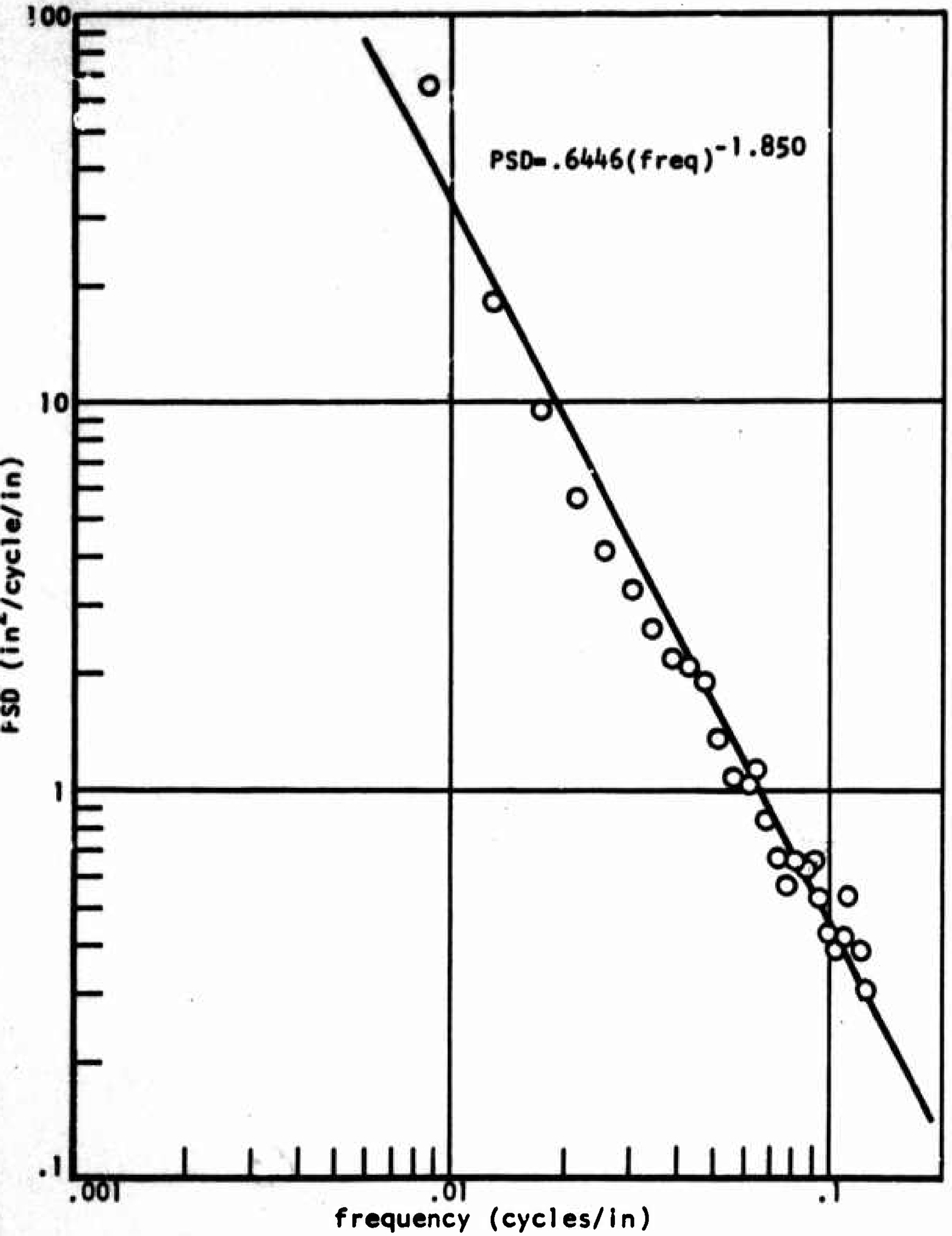


Figure 15. PSD of Profile 3

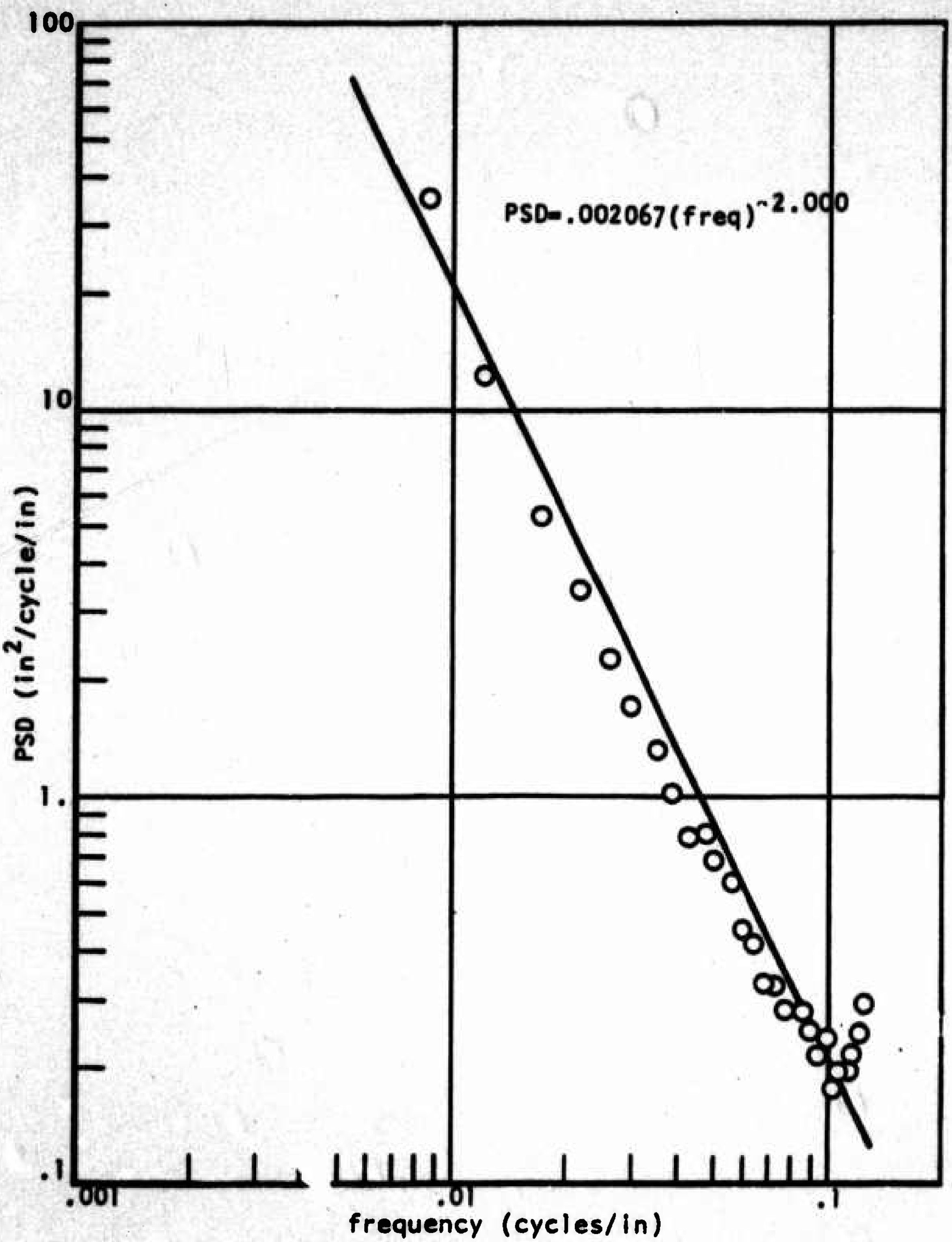


Figure 16. PSD of Profile 4

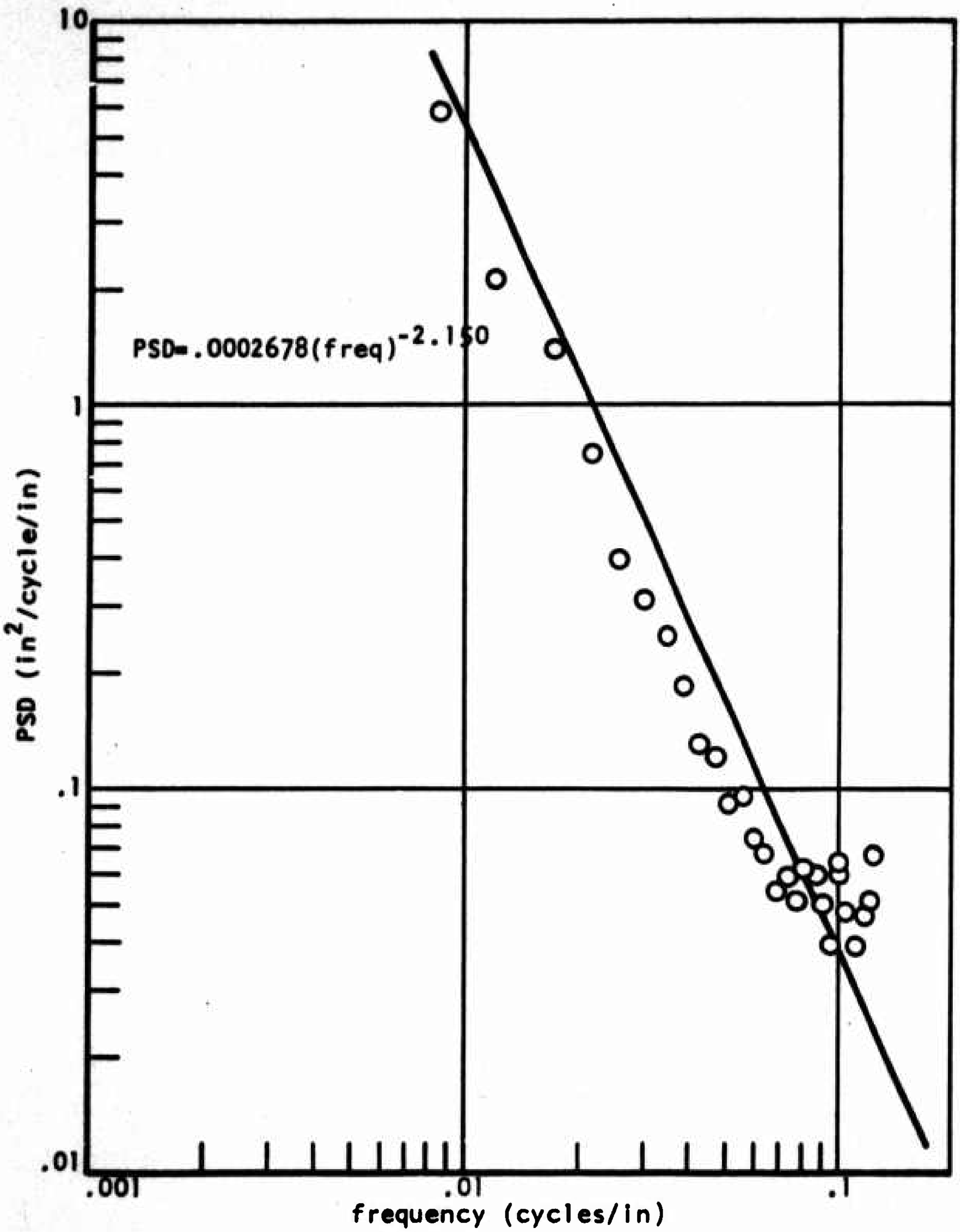


Figure 17. PSD of Profile 5

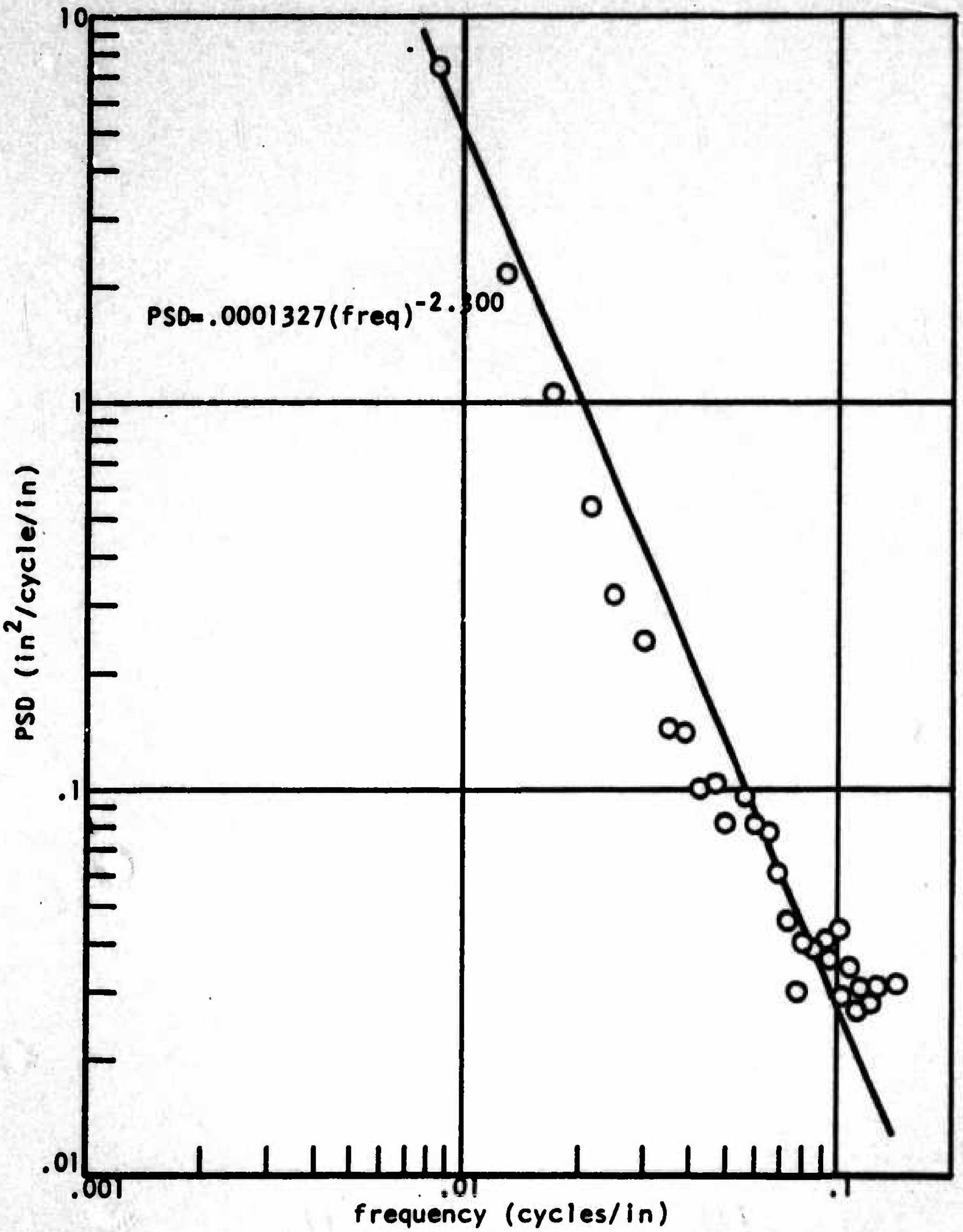


Figure 18. PSD of Profile 6

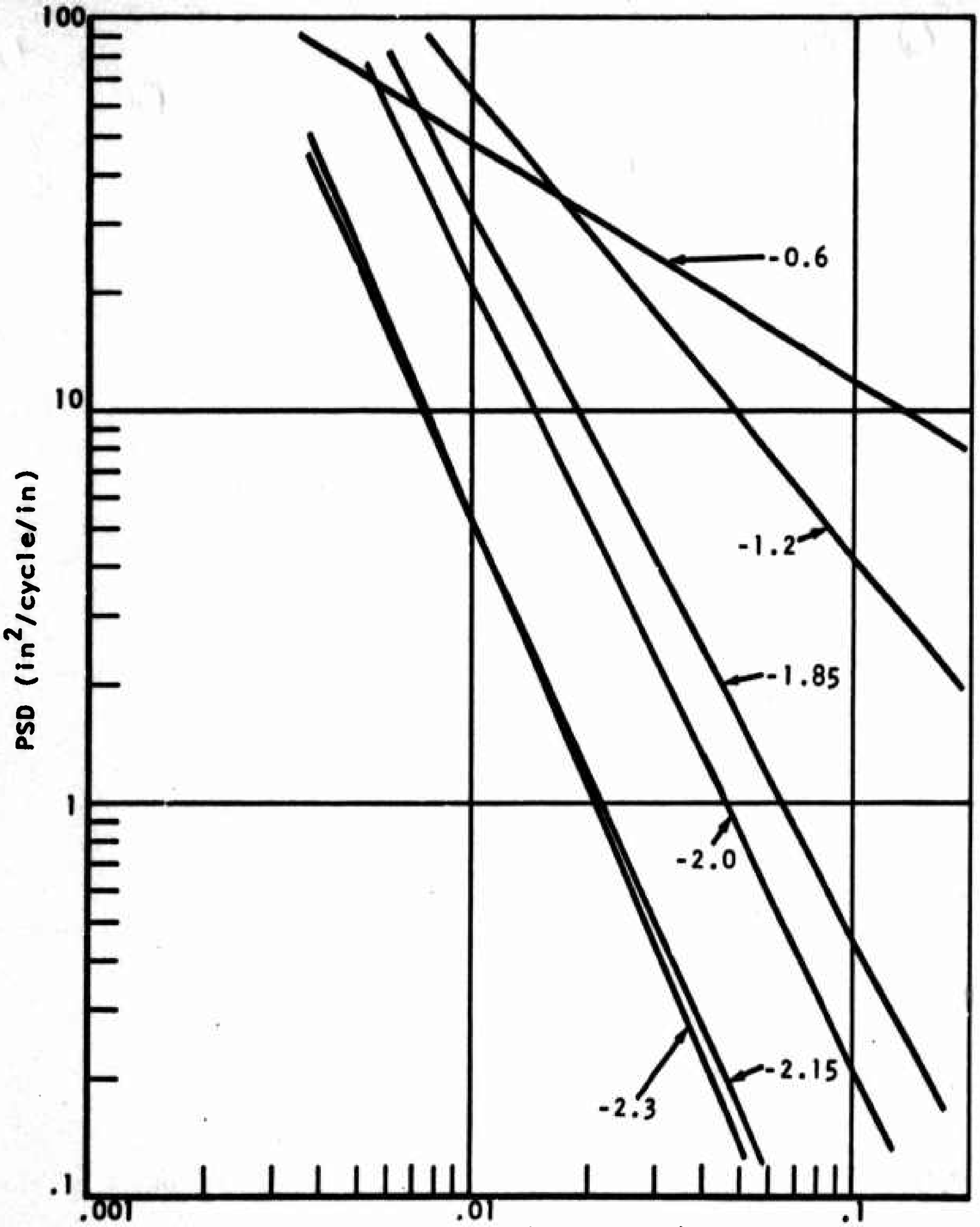


Figure 19. Comparison of PSD Curves

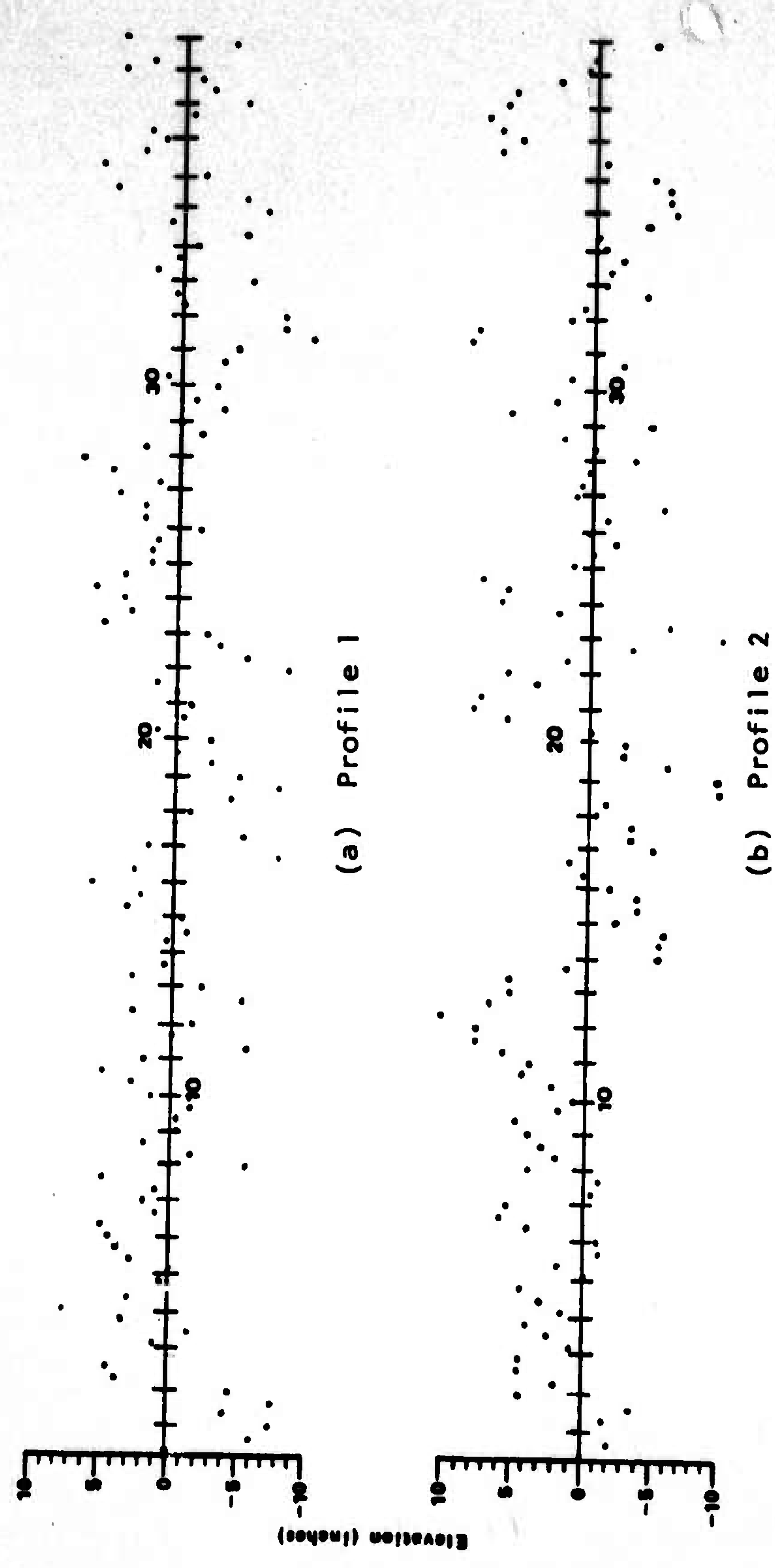


Figure 20. Section of Profiles 1 and 2
(NOTE: horizontal scale in feet)

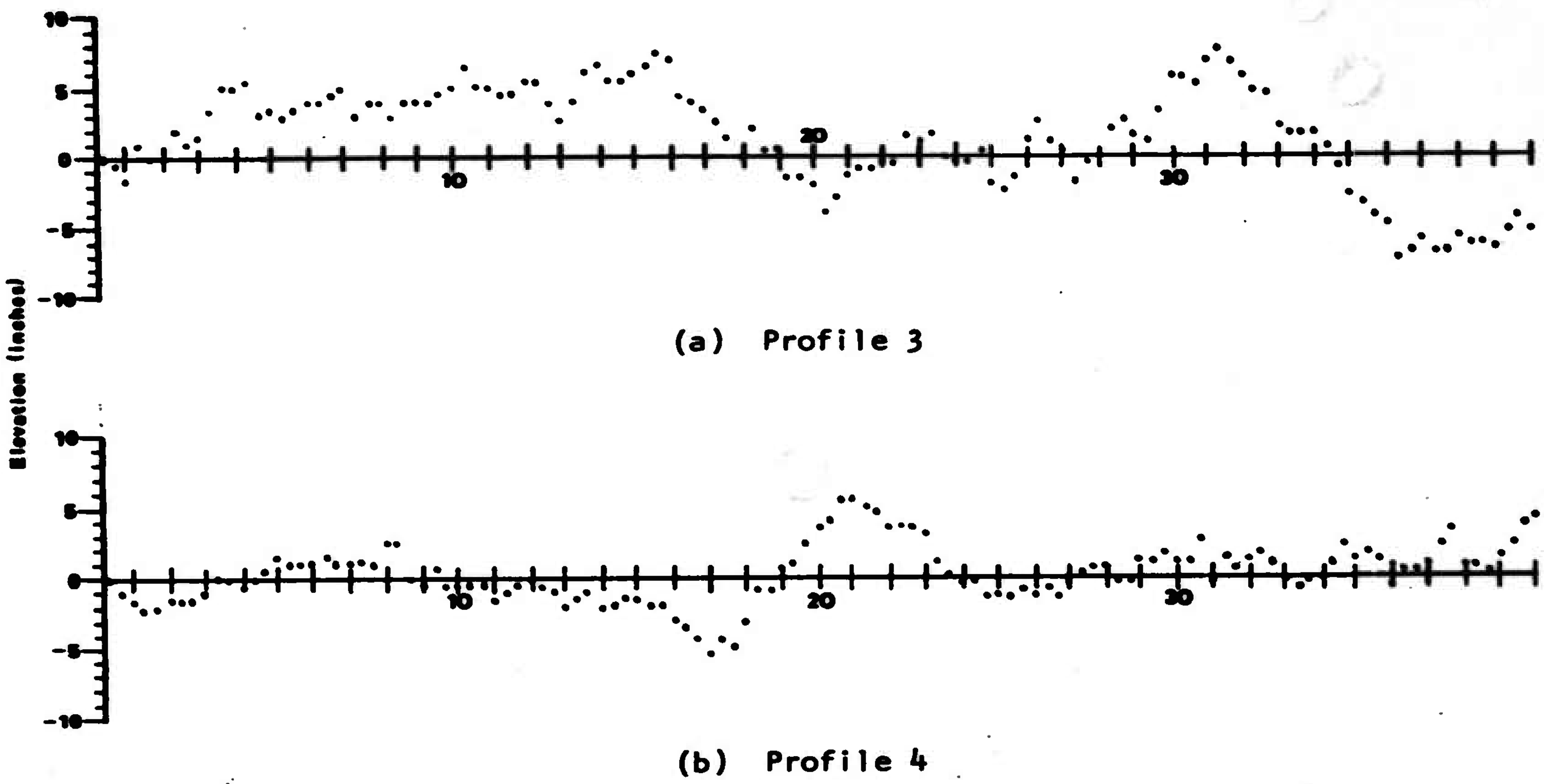
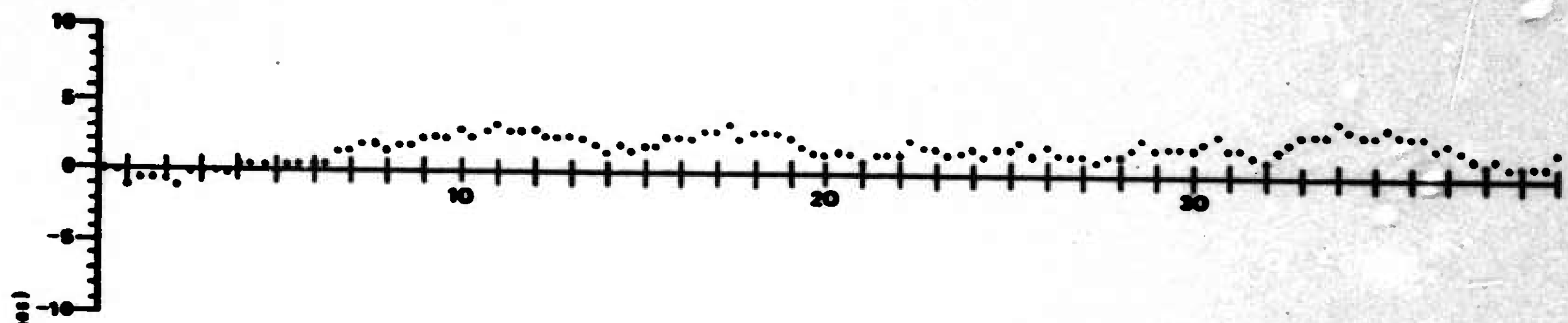
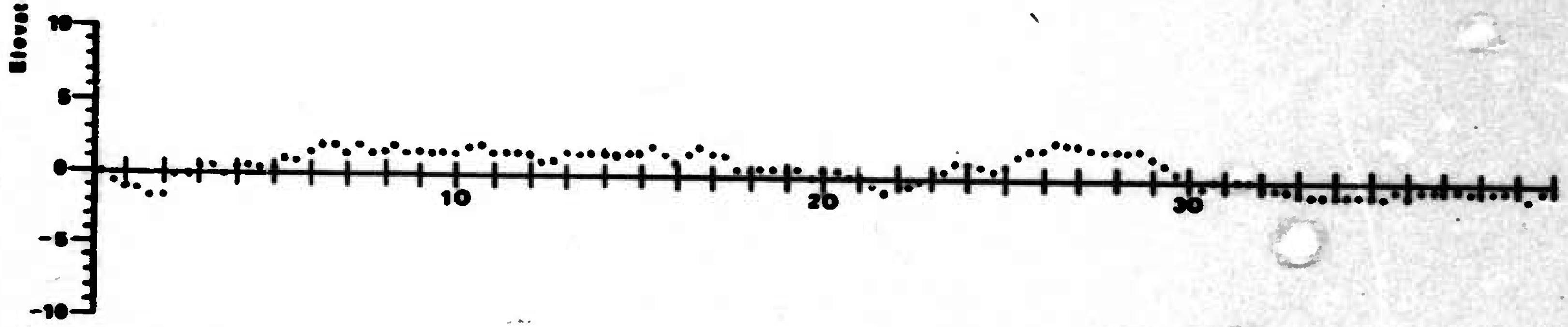


Figure 21. Section of Profiles 3 and 4
(NOTE: horizontal scale in feet)



(a) Profile 5



(b) Profile 6

Figure 22. Section of Profiles 5 and 6
(NOTE: horizontal scale in feet)

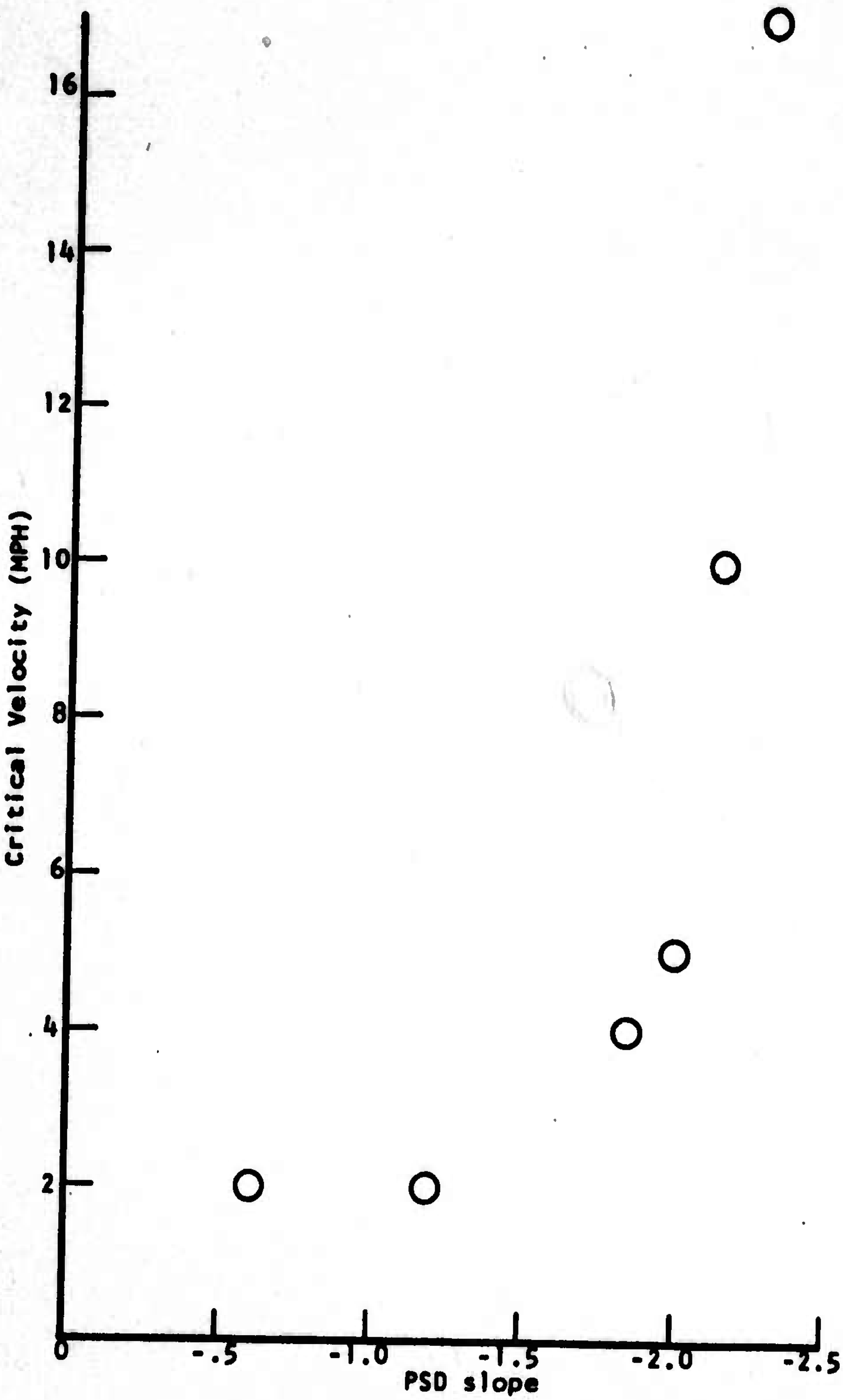


Figure 23. Results of M151
Critical Velocity Tests

however, the absorbed power jumped up sharply. To determine what may have happened, the M151 was exercised over one entire terrain (slope of -2.3) at a velocity of eleven miles per hour. The absorbed power was plotted using a special computer program, PWRPLT (see Appendix D). For this execution the absorbed power leveled off at a value of 5.3 watts after about 18 seconds of real time, as compared to where it began to level off at 3.4 watts around 6 seconds. Thus, the critical speeds determined for the M151 indicate the sensitivity of the simulation but can be assumed to be about 50 percent too high.

Since it took four or five runs to find each critical speed, it was determined that to continue the "critical speed" approach would increase the cost of the program by a factor of from 3 to 5. Since the necessary funds were not available, a slightly different approach was taken.

First, each vehicle was exercised over each terrain profile at 5 mph and at 18 mph. These two speeds are near the relative minimum and maximum of off-road mobility expectations and should exercise each vehicle in most of its tolerance range.

Second, instead of measuring discrete absorbed power values, a new subprogram, AVERAGE was written to compute a running average and PWRPLT was used to plot the average absorbed power.

Each vehicle-profile combination underwent a 15-20 second simulation. The final average absorbed power was used to determine simulation sensitivity to the PSD slope of the input terrain profile.

The results are given in Figures 24 and 25.

Unfortunately, at both 5 mph and 18 mph on the terrains with PSD slopes of -0.6 and -1.2, both vehicles exhibited large pitch variations. Since a small pitch angle approximation was used to derive the vehicle simulation, all data on those profiles was considered invalid and is not included in the analysis.

There is no scientific reason to draw a straight line between the two points at each slope. In fact, according to Murphy's data,[1] the lines should probably be concave downward (second derivative of the power with respect to slope negative). The lines are drawn simply to enhance visual comparison. Even a hasty inspection makes one thing obvious -- the vehicle model is extremely sensitive to the PSD slope of the terrain profile input. Of significant interest is the sensitivity in the region from -2.0 to -2.15, since many natural terrains fall into this region. The overlapping of the lower two lines is probably due to statistical scatter in the profile generation process.

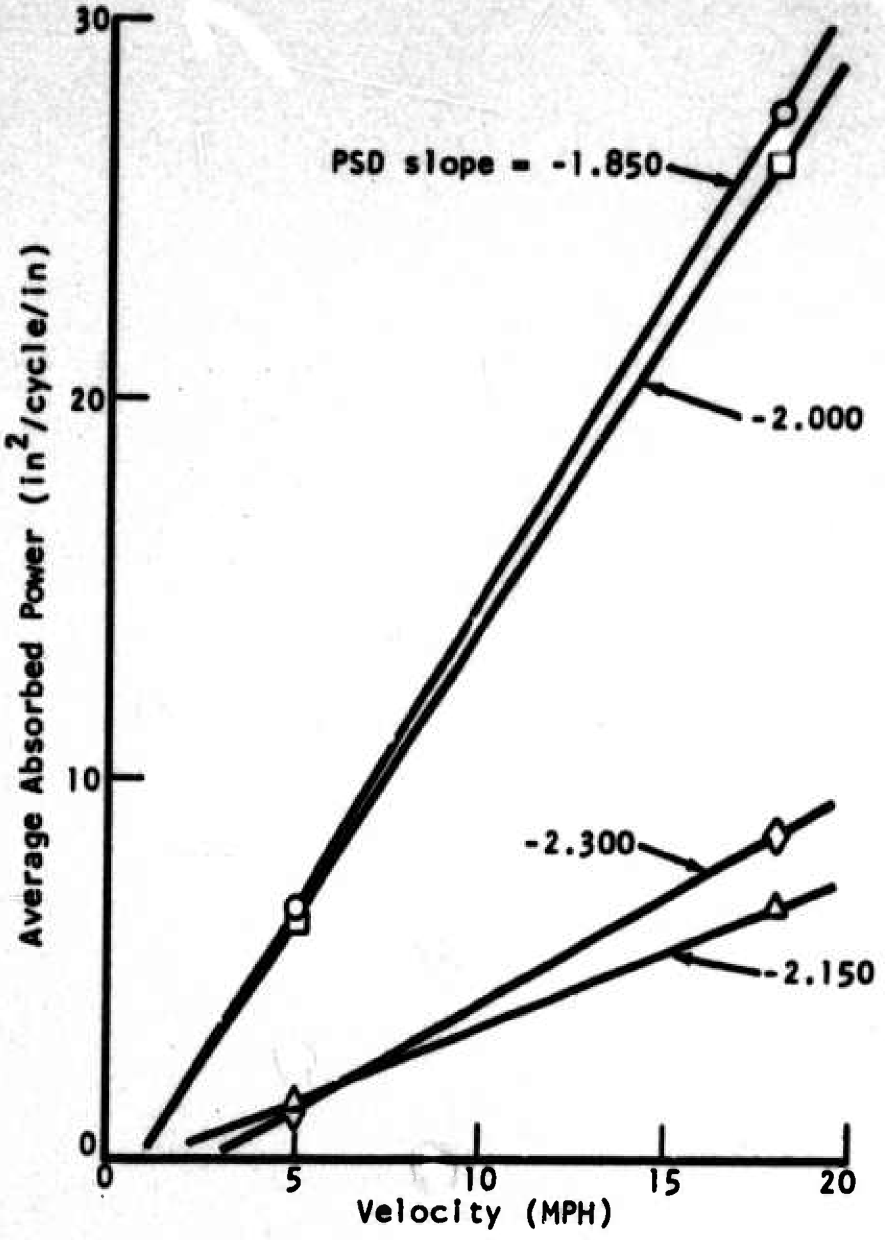


Figure 24. Results of M151 Average Absorbed Power Tests

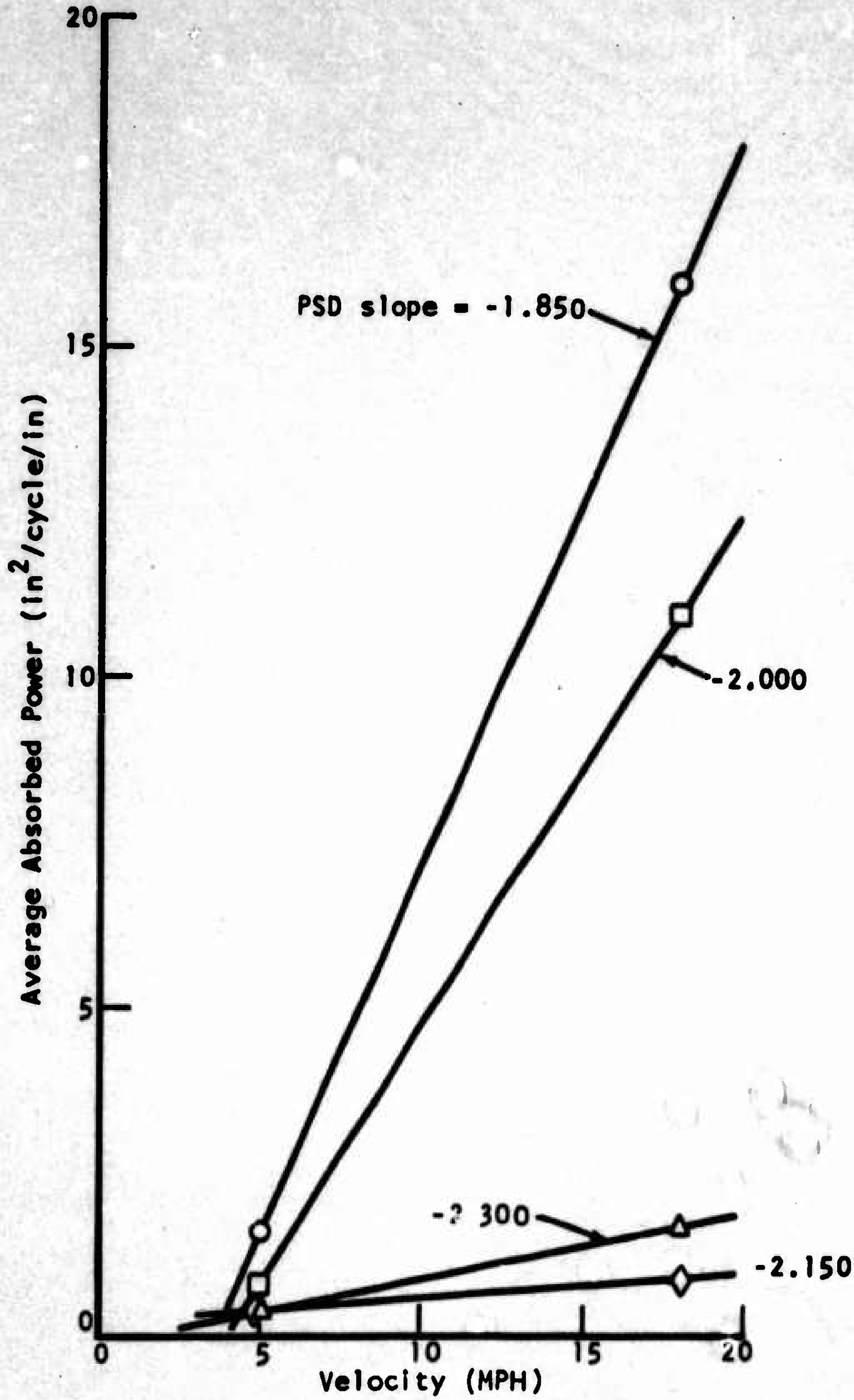


Figure 25. Results of M35 Average Absorbed Power Tests

II. The second objective of this research was to study the sensitivity of the simulation to changes in profile spacing. One random profile was created using NOIPSD with a profile spacing of 0.5 inch. One may think of this profile as constant, as if it existed in Nature somewhere. As the profile spacing (measurement interval) is changed, the profile itself will not be altered. What will change is the distance between points which are used as input to the vehicle simulation. The first few feet are shown in Figure 26(a). Since straight lines are used to interpolate between points in the vehicle simulation, they are drawn between them in the figure. If a one inch interval is desired, every other point is "read" by the computer and the result is a profile as in Figure 26 (b). The measurement interval is increased by .5 inch increments in Figures 26 (b-j). Note that as the measurement interval increases, the profile appears to become smoother as the valleys are bridged and the peaks flattened by the linear interpolation scheme. The 1/2-inch-interval points are repeated in Figure 26 (g) for comparison. Note how the first valley has been gradually disappearing and the peak is completely flat.

For this test, an RMS elevation of 1.0 inch was chosen in order to depict a more natural terrain. A PSD of slope -1.850 was specified to keep away from the 2.000-2.150 instability region. The desired profile was obtained from NOIPSD, stipulating NG = 7, N = 1800, RMS = 1.0 and γ = .046 in the input phase. The final PSD equation was PSD = .001134 $(\text{freq})^{-1.850}$ (see Figure 27).

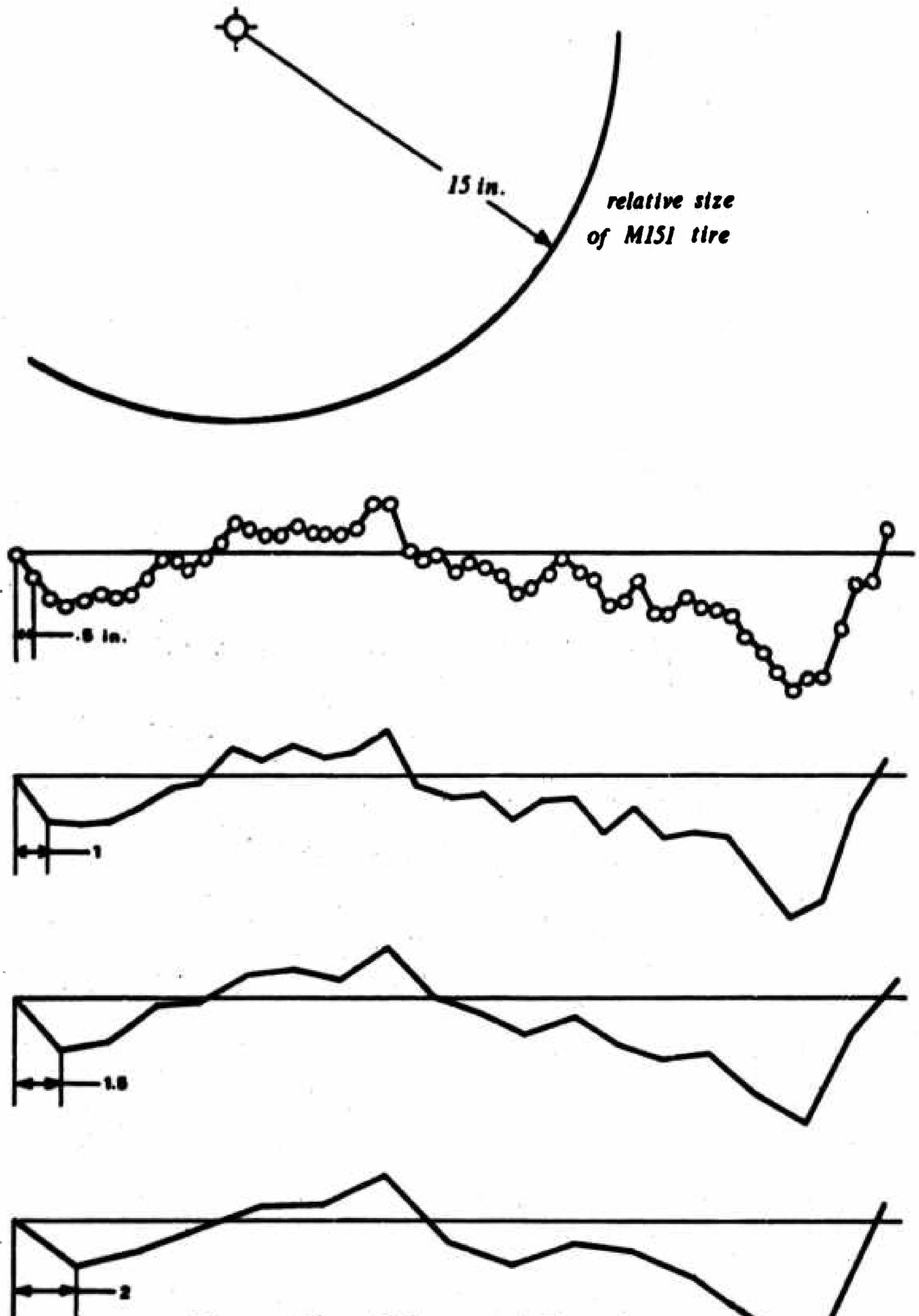


Figure 26. Effects of Changing
Input Profile Spacing

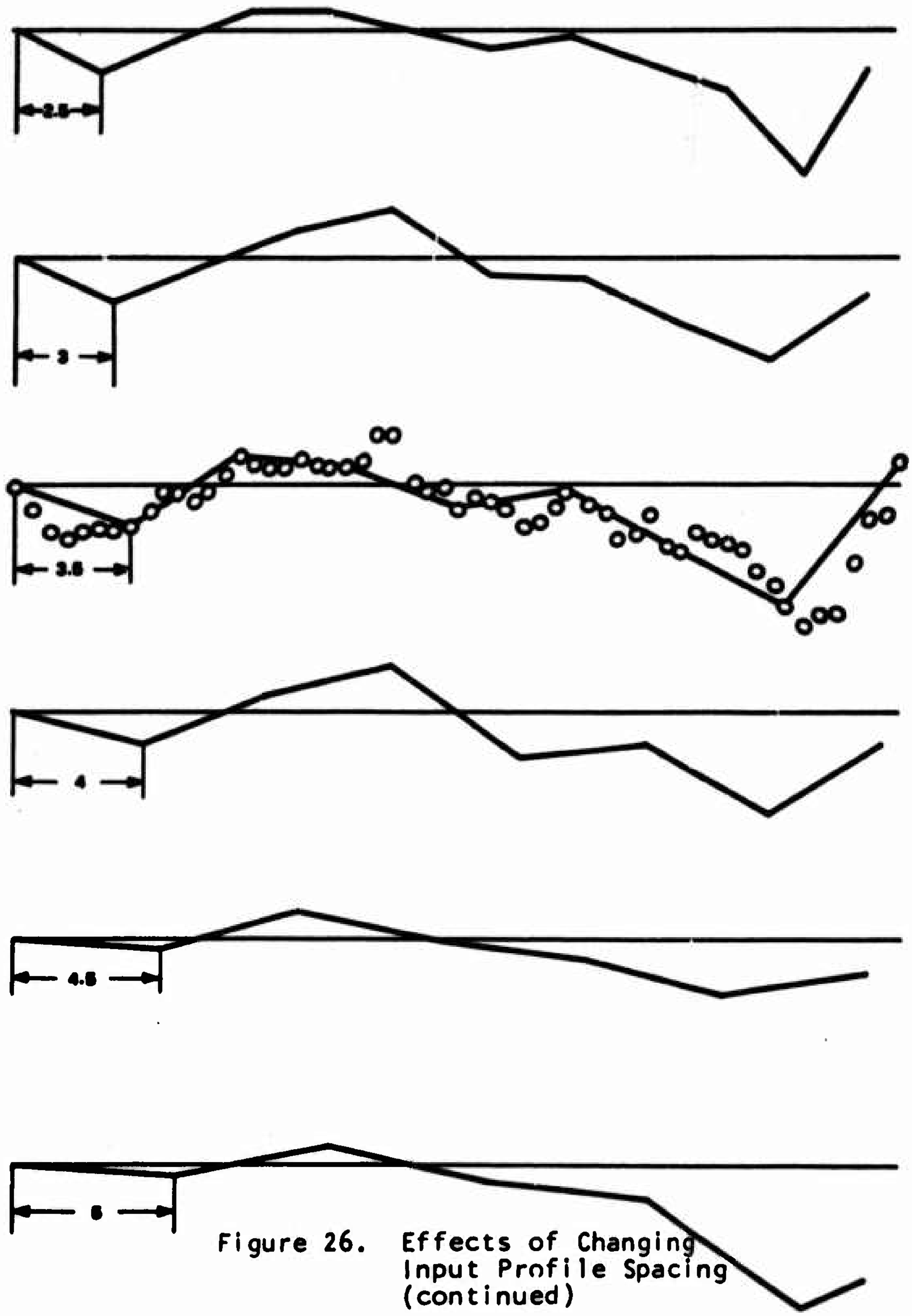


Figure 26. Effects of Changing Input Profile Spacing (continued)

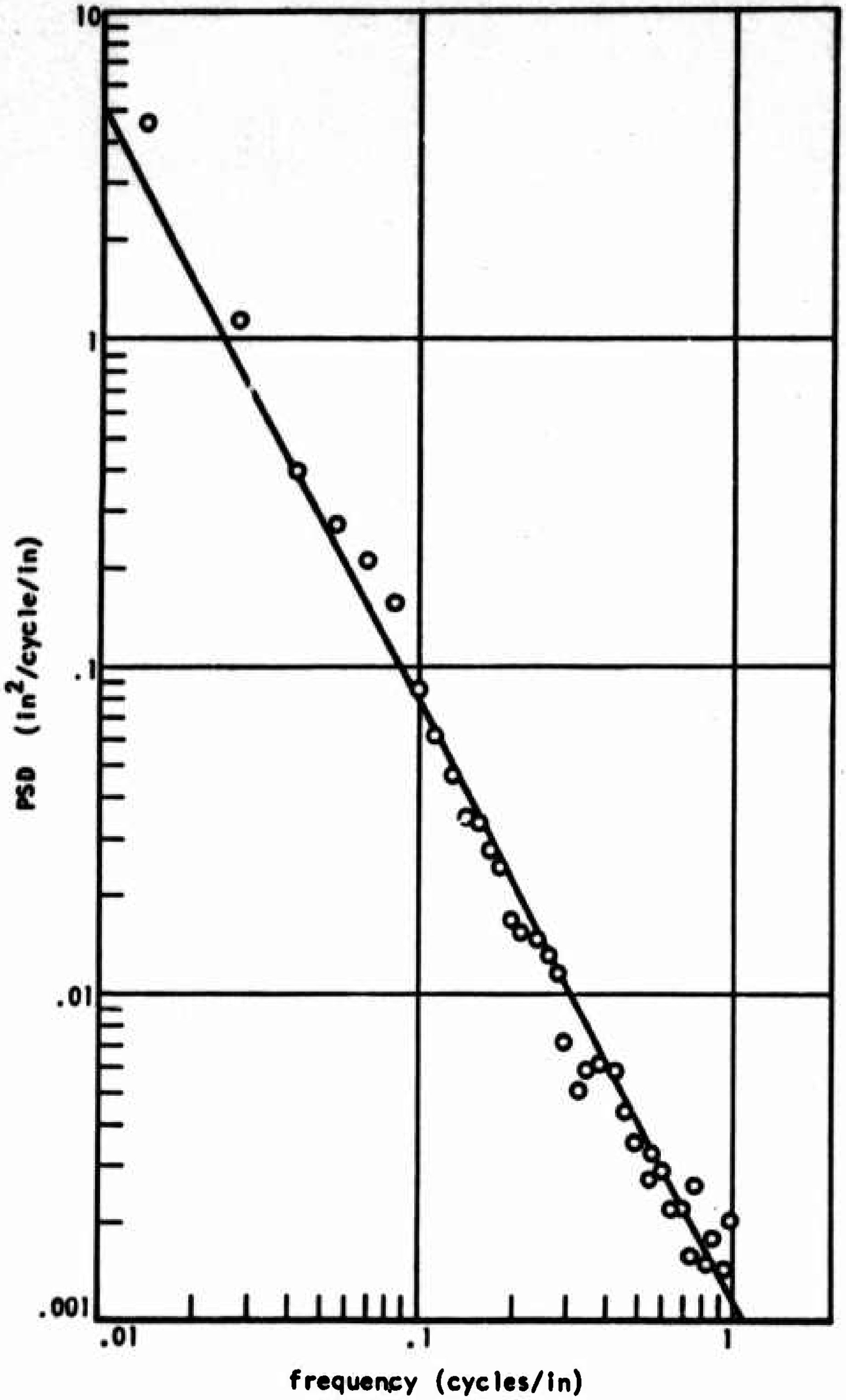


Figure 27. PSD of Profile Used for Measurement Interval Tests

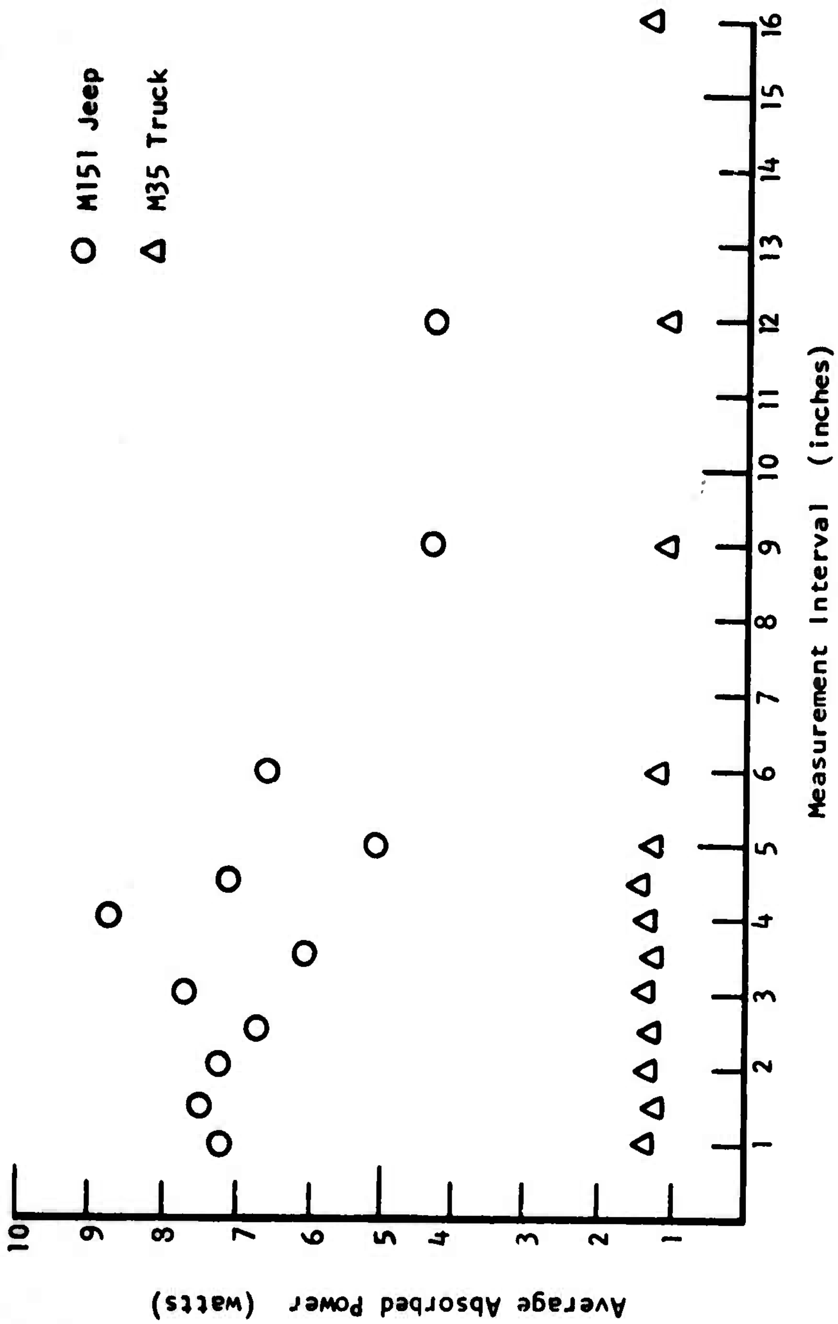
Each vehicle simulation was then exercised over the profile at different measurement intervals. Velocity was maintained at ten miles per hour. The simulation was exercised until a steady-state absorbed power level was reached and the average absorbed power was used to compare results. Table 4 and Figure 28 show the results.

Note that the average absorbed power for the M35 maintains an essentially constant level throughout the tested range of measurement interval. In fact, at sixteen inches, the tire model consists of only one vertical spring -- a "point follower." Thus it appears that for the M35 truck and probably for other heavy, long-wheelbase trucks, a "point follower" tire model is sufficient.

For the M151 Jeep, however, the plot shows a steady decrease in average absorbed power, after about 2.5 or 3.0 inches. For the M151 and probably for all short-wheelbase, small-tired vehicles, it is obvious that a smaller measurement interval is required to provide numerically stable absorbed power values. The data indicate that an interval of as low as 2.5 inches is necessary. For profiles with higher RMS elevations, the interval may be even smaller.

Table 4

Measurement Interval (Inches)	Average Absorbed Power (watts)	
	M151	M35
1.0	7.206	1.328
1.5	7.000	1.207
2.0	7.221	1.298
2.5	6.740	1.276
3.0	7.819	1.377
3.5	6.085	1.234
4.0	8.734	1.318
4.5	7.129	1.404
5.0	5.094	1.254
6.0	6.236	1.186
9.0	4.329	1.082
12.0	4.256	1.016
16.0	-	1.321



Fi.ure 28. Results of Profile Spacing Variation on Average Absorbed Power

CONCLUSIONS

1. The vehicle simulation is extremely sensitive to PSD slope, especially in the neighborhood of -2.0.

2. The vehicle simulation is sensitive to input profile spacing for light, short vehicles. A very small measurement interval of from 2.5 to 3 inches is necessary for numerically stable average absorbed power values.

3. The vehicle simulation is relatively insensitive to variations in measurement interval for heavy, large vehicles. Thus a much longer measurement interval is possible and a vastly simpler tire model can be used.

RECOMMENDATIONS

1. That the sensitivity of the model be further tested using terrains differing in PSD slope by much smaller increments between -1.85 (or even -1.70) and -2.30.

2. That the M35 simulation be further tested for sensitivity over terrain profiles with greater RMS elevations and different slopes.

REFERENCES

Footnoted:

1. Murphy, N. R., Jr., "Dynamics of Wheeled Vehicles, Report 3, A Statistical Analysis of Terrain-Vehicle-Speed Systems," Waterways Experiment Station, Vicksburg, Miss., April 1971.

2. Van Deusen, B. D., "A Study of the Vehicle Ride Dynamics Aspect of Ground Mobility, Vol. I, Summary" WES Contract Report No. 3-114, Detroit, Michigan, March 1965.

3. Lee, R. A. and Pradko, F., "Analytical Analysis of Human Vibration," SAE Paper 680091, 1968.

4. Pradko, F., Lee, R. A., and Greene, J. D., "Human Vibration Response Theory," ASME paper 65-WA/HUF-19, New York, 1965.

5. Kozin, F., Cote, L. J., Bogdanoff, J. L., "Statistical Studies of Stable Ground Roughness," USATAC Report No. 8391 (LL 95), Warren, Mich., November 1963.

6. Van Deusen, B. D., "A Statistical Technique for the Dynamic Analysis of Vehicles Traversing Rough Yielding and Non-Yielding Surfaces," NASA Contractor Report No. 659, Detroit, Mich., March 1967.

7. Blackman, R. B. and Tukey, J. W., "The Measurement of Power Spectra," Dover Publications Inc., New York, 1958.

8. "Ordnance Corps Equipment Data Sheets, TM 9-500," HQ, DA, Sept. 1962.

General:

Greenwood, D. T., Principles of Dynamics, Prentice-Hall, New Jersey, 1965.

Grether, W. F., "Vibration and Human Performance," Human Factors, John Hopkins Press, Maryland, 1971.

Holland, C. L., Jr., "Performance Effects of Long-Term Random Vertical Vibration," Human Factors, John Hopkins Press, Maryland, 1971.

Holland, C. L. Jr., Cote, L. J. and Bogdanoff, J. L., "Some Aspects of Human Response to Random Vibration," USATAC Report No. 9010 (LL 105), Warren, Michigan, August 1965.

Lee, R. A. and King, A. I., "Visual Vibration Response," ASME Paper 70-BHF-16, New York, 1970.

Lessem, A. S., "Dynamics of Wheeled Vehicles, Report 1, A Mathematical Model for the Traversal of Rigid Obstacles by a Pneumatic Tire," USAEWES, CE, Vicksburg, Miss., May 1968.

Lessem, A. S. and Murphy, N. R., Jr., "Studies of the Dynamics of Tracked Vehicles," USAEWES, Vicksburg, Miss., 1971.

McClellan, D. M., "Ride Response of a Model Vehicle to Harmonic Inputs," Report SIT-DL-71-1532, Stevens Institute of Technology, New Jersey, May 1971.

Shoenberger, R. W. and Harris, C. S., "Psychophysical Assessment of Whole-Body Vibration," Human Factors, John Hopkins Press, Maryland, 1971.

Timoshenko, S. and Young, D. H., Vibration Problems in Engineering, D. Van Nostrand Co., New Jersey, 1956.

APPENDIX A(Documentation of NOIPSD)

!. DESCRIPTION: This program is a combination of Murphy's two programs NOISE1 and PSD. It first generates, using Gaussian noise, a random terrain profile with zero mean and specified RMS elevation.

II. INPUTS AND OPERATING INSTRUCTIONS:

A. Prior to execution:

1. The equation FLAG = 6*N/(TAU*62) is written assuming 20 equivalent degrees of freedom. It comes from Blackman and Tukey, sections B.23 and B.24.[4]

2. The PSD estimates are smoothed by hamming. If other smoothing is desired, the appropriate parts of the program must be changed.

B. During execution:

1. NG: May be any number, but to insure differences among profiles, a prime number should be used.

2. SHORT TYPE OUT: Answer YES or NO.

a. YES causes a reduction in program options, setting TAU = 4.0, RMS = 4.0, N= 1200, and no output files are created.

b. NO causes the program to run normally, with all options open.

3. In put of TAU, N, and RMS: If no new values are entered, the program will set them to 4.0, 1200, and 4.0 respectively.

4. ALPHA: This is the parameter used to adjust the slope of the PSD curve.

5. FILES? Answer YES or NO.

   a. YES causes the program to ask for file names, which are entered if required. If no entry is made for either, that file will not be opened.

   b. NO causes no files to be created.

6. FLOW = : Enter the desired low cut-off frequency. If zero is entered, the program will set FLOW equal to .0002, which normally causes only the zero frequency estimate to be deleted.

III. OUTPUTS:

A. PSD file:

1. PSD equation in the form: $PSD = C(freq)^n$, where C is the intercept at frequency = 1.0 on the log-log curve. With PSD on the vertical axis, n is the slope of the curve -- a straight line.
2. RMS: This is the actual RMS as opposed to the desired RMS.
3. Power: The area under the PSD curve.
4. Number of Points: Number of points in profile.
5. Cut-off frequencies, low and high.
6. Points that were used in the square fit by sequence numbers.
7. List of frequencies and PSD estimates, five points to a line. Only the points used in the square fit are listed.

8. List of the points not used in the square fit.

B. Road file: Used as input to VEH.

1. The profile segment spacing in inches.
2. A line of identifying information.
3. The profile points in order, ten points per line.

C. Teletype:

1. RMS.
2. PSD equation. (If short type out is selected, only the slope of the PSD curve is printed out; none of the following are listed)
3. RMS, power, and number of points fitted by the above equation.
4. SD. (Standard deviation of the points from zero)
5. XBARAS: Mean of the profile after shift.
6. File names for output data. (If -----.DAT is typed out, no file was created.)

IV. SAMPLE EXECUTION:

A. Normal:

```
.EXECUTE NOIPSD
LOADING

LOADER 9K CORE
EXECUTION

NG = 7

SHORT TYPE OUT? NO

TAU=4.0,N=1200,RMS=4.0 UNLESS SPECIFIED NOW:
ALPHA = .0052
LLAG =  30
FILES? YES
PSD FILE NAME: PSD4
ROAD FILE NAME: ROAD4
AUTO-COVAR. IS COMPUTED
RMS =        4.000000
PSD IS FINISHED
FLOW = .0002

                                  -2.000
PSD =  0.2067366E-02 FREQ

RMS =   4.000000
POWER =   3.673574
EQN FITS PTS   2 THRU  30
SD =   0.6519456
XBARAS =   0.0000000
FILE FOR PSD PLOT: PSD4 .DAT
FILE FOR ROAD PLOT: ROAD4.DAT


EXECUTION TIME:          11.73 SEC.
TOTAL ELAPSED TIME:     2 MIN. 3.37 SEC.
NO EXECUTION ERRORS DETECTED

EXIT
```

B. Abbreviated:

```
.EXECUTE NOIPSD
LOADING

LOADER 9K CORE
EXECUTION

NG = 7

SHORT TYPE OUT? YES
ALPHA = .0052

SLOPE:-2.000


EXECUTION TIME:          6.40 SEC.
TOTAL ELAPSED TIME:     24.75 SEC.
NO EXECUTION ERRORS DETECTED

EXIT
```

V. PROGRAM LISTING:

```
      DIMENSION X(1200),DX(1200),FREQ(257)
      DIMENSION RX(250),PX(250),SPX(250)
      COMMON SD
      WRITE(6,900)
900   FORMAT(' NG = ',S)
      ACCEPT 997,NG
      TAU=0.
      N=0
      RMS=0.
      WRITE(6,901)
901   FORMAT(' SHORT TYPE OUT? ',S)
      ACCEPT 996, OPTYPE
      IF(OPTYPE.EQ.'YES')     GO TO 140
      WRITE(6,902)
902   FORMAT(' TAU=4.0,N=1200,RMS=4.0 UNLESS'
     +  ' SPECIFIED NOW!',S)
      ACCEPT 903, TAU,N,RMS
903   FORMAT(F,I,F)
140   IF(TAU.EQ.0.)     TAU=4.0
      IF(RMS.EQ.0.)     RMS=4.0
      IF(N.EQ.0)        N=1200
      GO TO 160
150   WRITE(6,907)
907   FORMAT('+ALPHA = ',S)
      ACCEPT 998, ALPHA
      IF(ALPHA.EQ.0.) GO TO 150
C *****
170   FHIGH=1./(2.*TAU)
      FLAG=6.*N/(TAU*62.)
      LLAG=IFIX(FLAG+.5)
      LAG=LLAG-1
      IF(OPTYPE.EQ.'YES')     GO TO 190
      WRITE(6,908) LLAG
908   FORMAT('+LLAG =',I4,/)
180   WRITE(6,909)
909   FORMAT('+FILES? ',S)
      ACCEPT 996,OPFIL
      IF(OPFIL.EQ.'NO') GO TO 190
      WRITE(6,910)
910   FORMAT('+PSD FILE NAME: ',S)
      ACCEPT 996, FN1
      IF(FN1.NE.' ')  CALL OFILE(21,FN1)
      WRITE(6,911)
911   FORMAT('+ROAD FILE NAME: ',S)
      ACCEPT 996, FN2
      IF(FN2.NE.' ')  CALL OFILE(22,FN2)
      GO TO 200
```

```
      190      FN1=' '
               FN2=' '
      C *****  SIGMAN AND ST. DEV. ARE COMPUTED
      200      SIGMAN=RMS*SQRT(1.-EXP(-2.*ALPHA*TAU))
               AA=EXP(-ALPHA*TAU)
               SD=SIGMAN*SIGMAN
      C *****  DISPLACEMENTS ARE OBTAINED
      C *****  FROM GAUSSIAN RANDOM NUMBERS
               SUM=0.
               DX(N)=0.
               NN=N-1
               SUM1=0.
               DO 210 I=1,NN
               CALL GAURND(V,NG)
               DX(I)=V
      210      SUM=SUM+DX(I)
               XBAR=SUM/NN
               DO 220 I=1,NN
               DX(I)=DX(I)-XBAR
      220      SUM1=SUM1+DX(I)
               XBARAS=SUM1/NN
               FACT=1.
               Y=0.
               X(1)=0.
               DO 230 I=2,N
      230      X(I)=DX(I)+X(I-1)*AA
      C *****  RMS IS COMPUTED AND ADJUSTED
      C *****  TO DESIRED LEVEL
      240      SUMX2=0.
               DO 250 I=1,N
               X(I)=FACT*X(I)
      250      SUMX2=SUMX2+X(I)*X(I)
               ARMS=SQRT(SUMX2/N)
               FACT=RMS/ARMS
               Y=Y+1.
               IF(Y.EQ.1.) GO TO 240
      C *****  AUTO-COVARIANCE COMPUTATION
      260      NPOINT=N
               DO 280 I=1,LLAG
               SX=0.
               M=I-1
               NX=NPOINT-M
               FNX=FLOAT(NX)
               DO 270 J=1,NX
               K=M+J
      270      SX=SX+X(J)*X(K)
      280      RX(I)=SX/FNX
               RMS=SQRT(RX(1))
               IF(OPTYPE.NE.'YES')      WRITE(6,912) RMS
        912    FORMAT('+AUTO-COVAR. IS COMPUTED'/,1X,
               'RMS = ',G18.7/)
```

```
      C ***** POWER SPECTRAL DENSITY COMPUTATION
              PI=3.14159265
              VV=PI/LAG
              W=2.*TAU/PI
              DELF=1/(2*LAG*TAU)
              RX(1)=.5*RX(1)
              RX(LLAG)=.5*RX(LLAG)
              DO 300 IH=1,LLAG
              SC=0.
              DO 290 IP=1,LLAG
              U=(IH-1)*(IP-1)*VV
        290   SC=SC+RX(IP)*COS(U)
              PX(IH)=W*SC
        300   FREQ(IH)=(IH-1)*DELF
              SPX(1)=.54*PX(1)+.46*PX(2)
              SPX(LLAG)=.54*PX(LLAG)+.46*PX(LLAG-1)
              KK=LLAG-1
              IF(SPX(1).LT.0.) SPX(1)=0.
              DO 310 J=2,KK
              SPX(J)=.54*PX(J)+.23*(PX(J+1)+PX(J-1))
        310   IF(SPX(J).LT.0.) SPX(J)=0.
              IF(OPTYPE.NE.'YES')     WRITE(6,913)
          913 FORMAT('+PSD IS FINISHED'/)
      C ***** SQUARE-FIT ROUTINE
              FLOW=0.
              NS=0
              NF=LLAG
              IF(OPTYPE.NE.'YES')     WRITE(6,914)
          914 FORMAT('+FLOW = ',$)
              IF(OPTYPE.NE.'YES')     ACCEPT 998,FLOW
              IF(FLOW.EQ.0.) FLOW=.0002
              DO 320 I=1,LLAG
              IF(FREQ(I).LT.FLOW) NS=NS+1
        320   IF(FREQ(I).GT.FHIGH) NF=NF-1
              RX(1)=0.
              PX(1)=ALOG10(SPX(1))
              NP=NF-NS
              NS=NS+1
              DO 340 I=NS,NF
              IF(SPX(I).LT..0000005)  GO TO 330
              RX(I)=ALOG10(FREQ(I))
              PX(I)=ALOG10(SPX(I))
              GO TO 340
        330   PX(I)=0.
              RX(I)=0.
              NP=NP-1
        340   CONTINUE
```

```
C *****
      SUMY=0.
      SUMX=0.
      SUMXY=0.
      SUMX2=0.
      DO 350 I=NS,NF
      SUMY=SUMY+PX(I)
      SUMX=SUMX+RX(I)
      SUMXY=SUMXY+(PX(I)*RX(I))
      SUMX2=SUMX2+(RX(I)**2.)
350   CONTINUE
      DE=NP*SUMX2-SUMX*SUMX
      AM=(NP*SUMXY-SUMX*SUMY)/DE
      B= (SUMX2*SUMY-SUMX*SUMXY)/DE
      P=10.**B
C *****
      PSUM=0.
      DO 360 I=1,LLAG
360   PSUM=PSUM+SPX(I)*DELF
C ***** THE DATA IS NOW OUTPUT TO FILES
C ***** AND TTY AS DESIRED
      IF(OPTYPE.EQ.'YES')     WRITE(6,915) AM
915   FORMAT(' SLOPE=',F6.3/)
      IF(OPTYPE.NE.'YES')  WRITE(6,916) AM,B,ARMS,
     & PSUM,NS,NF
916   FORMAT(27X,F6.3,/' PSD =',
     + G,' FREQ'//,' RMS =',G,/' POWER =',G/,
     + ' EQN FITS PTS',I4,' THRU',I4)
      IF(FN1.EQ.' ') GO TO 480
      WRITE(21,920) AM,B,PSUM,FLOW,FHIGH,NS,NF
920   FORMAT(45X' ***** POWER SPECTRAL'
     + ' DENSITY EQUATION *****'/
     + 45X,' *',41X,'*'/
     + 45X,' *',31X,F6.3,4X,'*'/
     + 45X,' *',5X,'PSD = ',G,' FREQ  ',8X,'*'/
     + 45X,' *',41X,'*'/,
     + 45X,' ************************'
     + '*********************'//,
     + 61X,'POWER = ',F10.4,/
     + 50X,'CUTOFF FREQUENCIES!',/
     + 50X,'     LOW-- ',F10.8,/
     + 50X,'    HIGH-- ',F10.8,/
     + 50X,'EQN FITS POINTS',I4,' THRU',I4,//)
      WRITE(21,921)
921   FORMAT(' THE FOLLOWING POINTS WERE USED'
     + ' IN THE SQ FIT!'/)
      NS=NS-1
      WRITE(21,922) ((FREQ(K),SPX(K)),K=NS,NF)
922   FORMAT(5(F18.8,F8.3)/)
```

```
        MF=NF+1
        WRITE(21,924)
  924   FORMAT(///' THE FOLLOWING POINTS WERE'
     +  ' NOT FITTED:'/)
        IF(NS.EQ.1) GO TO 410
        WRITE(21,922) ((FREQ(K),SPX(K)),K=1,NS-1)
        GO TO 420
410     WRITE(21,925)
  925   FORMAT(' NO LOWER FREQUENCIES WERE DELETED'/)
420     IF(NF.EQ.LLAG) GO TO 460
        WRITE(21,922) ((FREQ(K),SPX(K)),K=MF,LLAG)
        GO TO 470
460     WRITE(21,926)
  926   FORMAT(/' NO HIGHER FREQUENCIES
        WERE DELETED'//)
470     CONTINUE
C *****
480     IF(FN1.EQ.' ') FN1='-----'
        IF(FN2.EQ.' ') FN2='-----'
        IF(OPTYPE.NE.'YES')  WRITE(6,927) SD,
     &  XBARAS,FN1,FN2
  927   FORMAT(' SD =',F12.7,/' XBARAS =',
     +  F12.7,/' FILE FOR PSD PLOT: ',A5,'.DAT',/
     +  ' FILE FOR ROAD PLOT: ',A5,'.DAT'/)
        IF(FN1.EQ.'-----')  GO TO 490
        WRITE(21,928) ARMS,SD,XBAR,XBARAS,NG,N
  928   FORMAT(67X,'RMS  =',F12.7,/
     +  53X,'STANDARD DEVIATION =',F12.7,/
     +  54X,'X-BAR AFTER GAUSS =',F12.7,/
     +  54X,'X-BAR AFTER SHIFT =',F12.7,/
     +  45X,'STARTING NUMBER FOR RAN(Z) =',I4,/
     +  55X,'NUMBER OF POINTS =',I6)
  929   FORMAT(10F12.8)
490     IF(FN2.EQ.'-----') GO TO 9999
        WRITE(22,930) TAU,FN2,AM,NG,ALPHA,TAU,RMS
  930   FORMAT(1X,F/1X,A5,'.DAT--SLOPE=',F6.3,
     +  ' NG=',I2,' ALPHA=',F6.5,
     +  ' TAU=',F3.1,' RMS=',F3.1)
        DO 500 I=10,N,10
500     WRITE(22,929) (X(K),K=I-9,I)
 9999   CALL EXIT
  996   FORMAT(A5)
  997   FORMAT(I)
  998   FORMAT(F)
  999   FORMAT(/)
        END
```

```
      SUBROUTINE GAURND(V,NG)
      COMMON SD
      V=0.
      J=0
      DO 100 I=1,NG
100   A=RAN(Z)
      DO 110 I=1,12
      A=RAN(Z)
110   V=V+A
      V=V-6.
      V=V*SD
      RETURN
```

APPENDIX B(Documentation for VEH)

I. DESCRIPTION: This program, in its original form, was also obtained from N. R. Murphy, WES. It is a non-linear, time-domain wheeled vehicle simulation on a digital computer, a PDP 10.

II. INPUTS AND OPERATING INSTRUCTIONS:

A. Prior to execution: The vehicle constants must be entered in Subroutine DATA (see Appendix C) and the tire load-deflection curves must be on hand.

B. During execution:

1. Name of input file (Road file created with NOIPSD).

2. Desired Delta-L: This should be a multiple of the input profile spacing.

3. Name of vehicle to be simulated.

4. Options. Answer YES or NO.

a. Absorbed Power.

b. Detailed output file?

(1) YES causes program to ask for file name, which should not be more than five characters long.

(2) NO causes program to skip the above question and no output file will be created.

c. Peak accelerations.

d. Driver motions (only asked if vehicle is a track)

e. RMS of all accelerations. (This must be answered YES if PWRPLT is to be used later to plot the

absorbed power.

f. Tire deflections. The program will type out the loads on each axle, front to rear. Enter the deflection caused by each load on a single tire.

5. Vehicle velocity in miles per hour.

6. Teletype printout time interval in seconds.

7. Time and absorbed power only? Anser YES or NO.

   a. YES causes the program to type out only time, absorbed power, and average absorbed power for each interval.

   b. NO will cause the normal printout format to be used.

8. Stop? Answer YES or NO. Program will stop on entry of YES.

III. OUTPUTS: These depend on the options selected. Assuming all are desired, the outputs will be as follows:

A. To detailed file:

1. List of THRESH and GAMMA.
2. Vehicle velocity in MPH and inches per second.
3. Delta-L.
4. Delta-t.
5. Number of steps in RKG integration.
6. RKG step size.
7. The input profile identification.
8. Time, profile input point, average power, absorbed power, and vehicle motions at each step.

B. To teletype: Basically the same information as above is printed out on the teletype, except that the time, input point, etc. are printed out only according to the teletype print-out interval specified earlier.

IV. SAMPLE EXECUTION:

IV. SAMPLE EXECUTION:

A. Normal:

```
.EXECUTE VEH
FORTRAN:  VEH
LOADING

LOADER 9K CORE
EXECUTION

FILE NAME OF INPUT PROFILE: ROADX
 DESIRED DELTAL: 1.
NAME OF VEHICLE? M151

DO YOU WANT THE FOLLOWING OPTIONS?

ABSORBED POWER? YES
A DETAILED OUTPUT FILE? YES
   FILE NAME: FILEX
PEAK ACCELERATIONS? NO
RMS OF ALL ACCELS? YES

FROM FORCE-DEFL CURVES FOR
M-151 JEEP TIRES, ENTER DEFLECTIONS:
  LOAD =  581.071   .82
  LOAD =  623.249   .865


VEHICLE VELOCITY IN MPH: 10.0
TTY PRINTOUT TIME INTERVAL: .2

TIME & POWER TYPEOUT ONLY? NO



VELOCITY=10.00 MPH (176.00 IPS)
DELTA-L=1.000   DELTA-T=0.0057
NSTEPS=   5    H=.001136
VEHICLE IS: M-151 JEEP
INPUT PROFILE IS:
ROADX.DAT--FILE FOR VARIABLE MEASUREMENT INTERVALS
```

A. Normal: (continued)

```
                DISPL      VELOC      ACCEL      RMSAC

TIME= 0.000   INPUT=  0.000   ABSORBED POWER=  0.000   0.000

C-G       -1.17601    0.00000    0.00000    0.00000
PITCH     -0.00086    0.00000    0.00000    0.00000
AXLE1     -0.82000    0.00000    0.00000    0.00000
AXLE2     -0.86500    0.00000    0.00000    0.00000
STOP? NO

TIME= 0.205   INPUT= -0.209   ABSORBED POWER=  0.140   0.041

C-G       -1.22561   -3.25482   -0.21239    0.15562
PITCH     -0.00026   -0.13723   -2.62326    2.13225
AXLE1     -1.06928   -8.41071    1.23916    0.45973
AXLE2     -0.86037    1.79340    0.04520    0.09609
STOP? NO

TIME= 0.403   INPUT= -0.549   ABSORBED POWER=  0.515   0.231

C-G       -1.09423   -1.51579    3.12303    0.23038
PITCH      0.00089   -0.05403    2.84386    3.34899
AXLE1     -0.72493   -0.32713    0.13655    0.65754
AXLE2     -0.80245    0.23447   -0.15274    0.12382
STOP? YES

EXECUTION TIME:          29.27 SEC.
TOTAL ELAPSED TIME:      3 MIN. 57.30 SEC.
NO EXECUTION ERRORS DETECTED

EXIT
```

B. Abbreviated:

```
.EXECUTE VEH
LOADING

LOADER 9K CORE
EXECUTION

FILE NAME OF INPUT PROFILE: ROADX
 DESIRED DELTAL: 1.5
NAME OF VEHICLE? M151

DO YOU WANT THE FOLLOWING OPTIONS?

ABSORBED POWER? YES
A DETAILED OUTPUT FILE? NO
PEAK ACCELERATIONS? NO
RMS OF ALL ACCELS? NO

FROM FORCE-DEFL CURVES FOR
M-151 JEEP TIRES, ENTER DEFLECTIONS:
  LOAD =  581.071   .82
  LOAD =  623.249   .865


VEHICLE VELOCITY IN MPH: 10.0
TTY PRINTOUT TIME INTERVAL: .1

TIME & POWER TYPEOUT ONLY? YES


 TIME  ABSPWR  AVEPWR
 0.00   0.00    0.00
 0.10   0.05    0.01
 0.20   0.19    0.04
 0.31   0.73    0.18
 0.40   0.78    0.30
 0.50   0.61    0.38
 0.61   0.88    0.43
 0.71   0.75    0.50
 0.80   0.89    0.52
 0.90   1.84    0.64
STOP? YES


EXECUTION TIME:          39.78 SEC.
TOTAL ELAPSED TIME:      1 MIN. 57.87 SEC.
NO EXECUTION ERRORS DETECTED
```

V. PROGRAM LISTING:

```
      C ***** STORAGE ALLOCATIONS
      C    NOTE: THE FOLLOWING COMMON STATEMENTS
      C       MUST APPEAR IN EACH SUBROUTINE
              COMMON FORCH(6),FORCT(7),FORCK(6),FORCH(6)
              COMMON SPDEF(6),DSPSF(6),THRESH(200),SIGMA(9)
              COMMON VAR(18),Y(260),PWRVAR(9),DAMP(6)
              COMMON ACCISS(9),ACCGS(9),ACCMAX(9),ACCMIN(9)
              COMMON SUMRMS(9),RMS(9),LEN(10),MASS(6)
              COMMON H,T,DELTAT,DELTAL,VELIPS,VELMPH,NSTEPS
              COMMON YIN,DRVMAX,DRVMIN,ABSPWR,GAMMA(200)
              COMMON DISDRV,VELDRV,ACCDRV,RMSDRV
              COMMON IFPWR,IFFILE,IFPACC,IFDRV,IFRMS
              COMMON NY,IDF,NAXLES,NSEGS,IFHORZ,FNAME
              COMMON FMASS,INRTIA,HORMOM,DRVLEN
              COMMON VEHQID(2),PROFIL(260),PASTP(260),INDEX
              COMMON YY(3),WEIGHT(3)
              COMMON SLIMIT(4,3),SSLOPE(5,3),SINT(5,3)
              COMMON DLIMIT(2,3),DSLOPE(3,3),DINT(3,3)
              DIMENSION DRIVER(4),IOPT(6),NYTEMP(3)
              DIMENSION FID(12),XTNAME(4)
              DIMENSION FK(18),P(18),Q(18),PY(9),PWRFK(9)
              DIMENSION PP(9),QQ(9)
              REAL LEN,MASS,INRTIA
              EQUIVALENCE (ISETUP,NY)
              EQUIVALENCE (DRIVER(1),DISDRV)
              EQUIVALENCE (IOPT(1),IFPWR)
              DATA XTNAME/'M151','M35',' ',' '/
              NO='N'
              IYES='Y'
              IFSTOP='N'
      C ***** VARIABLE INITIALIZATION
              DO 100 I=1,9
              PWRVAR(I)=0.
              QQ(I)=0.
              ACCISS(I)=0.
              SUMRMS(I)=0.
              Q(I)=0.
              ACCMAX(I)=0.
              ACCMIN(I)=0.
        100   CONTINUE
              ACCDRV=0.
              DO 110 I=1,18
        110   VAR(I)=0.
              T=0.
              SDVRMS=0.
              ABSPWR=0.
              DRVMAX=0.
              DRVMIN=0.
              YIN=0.
```

```
      H=.001
      DELTAL=4.
      JSTOP=1
      NSTOP=0
      TPRINT=0.
C ***** OPEN INPUT FILE AND READ PROFILE
C ***** SPACING AND OTHER IDENTIFICATION
      WRITE(6,900)
  900 FORMAT(' FILE NAME OF INPUT PROFILE: ',$)
      READ(5,996) FN2
      IF(FN2.EQ.' ') GO TO 120
      CALL IFILE(22,FN2)
      CALL FILIN(FID,JS,SPACING,1)
C ***** VEHICLE CONSTANTS READ-IN
  120 WRITE(6,902)
  902 FORMAT('+NAME OF VEHICLE? ',$)
      READ(5,996) TNAME
      DO 130 I=1,4
      IF(TNAME-XTNAME(I))130,140,130
  130 CONTINUE
      WRITE(6,903) XTNAME
  903 FORMAT(' THE AVAILABLE VEHICLES ARE:'
     + 4(2X,A5))
      GO TO 120
  140 GO TO (150,160,170,180) I
  150 ASSIGN 400 TO ISUB
      GO TO 190
  160 ASSIGN 410 TO ISUB
      GO TO 190
  170 ASSIGN 420 TO ISUB
      GO TO 190
  180 ASSIGN 430 TO ISUB
  190 CONTINUE
C ***** SELECTION OF OPTIONS
      WRITE(6,904)
  904 FORMAT(' DO YOU WANT THE FOLLOWING',
     + ' OPTIONS?'//' ABSORBED POWER? ',$)
      READ(5,995) IFPWR
      WRITE(6,905)
  905 FORMAT('+A DETAILED OUTPUT FILE? ',$)
      READ(5,995) IFFILE
      IF(IFFILE.EQ.'N')  GO TO 210
      WRITE(6,906)
  906 FORMAT('+    FILE NAME: ',$)
      ACCEPT 996, FN1
      CALL OFILE(21,FN1)
  210 WRITE(6,907)
  907 FORMAT('+PEAK ACCELERATIONS? ',$)
      READ(5,995) IFPACC
      IF(I.LT.3)  GO TO 215
```

```
          WRITE(6,908)
    908   FORMAT('+DRIVER MOTIONS? ',$)
          READ(5,995) IFDRV
          GO TO 216
  215     IFDRV='N'
  216     WRITE(6,909)
    909   FORMAT('+RMS OF ALL ACCELS? ',$)
          READ(5,995) IFRMS
          CALL DATA(I)
C *****  INITIALIZE PROFILE ARRAYS
          DO 220 I=1,NY
          PASTP(I)=0.
          Y(I)=0.
  220     PROFIL(I)=0.
          NYTEMP(1)=0.
          NYTEMP(2)=(LEN(1)+LEN(2)-LEN(3))/DELTAL
          NYTEMP(3)=(LEN(1)+LEN(2)+LEN(4))/DELTAL

C *****  VEHICLE RUN VARIABLE INPUT
          WRITE(6,911)
    911   FORMAT(/' VEHICLE VELOCITY IN MPH: ',$)
          READ(5,998) VELMPH
          WRITE(6,912)
    912   FORMAT('+TTY PRINTOUT TIME INTERVAL: ',$)
          READ(5,998) TIP
          WRITE(6,913)
    913   FORMAT(' TIME & POWER TYPEOUT ONLY? ',$)
          READ(5,995) OPTYPE
C *****  TIME STEP AND RKG TIME SET-UP
          VELIPS=VELMPH*17.6
          DELTAT=DELTAL/VELIPS
          NSTEPS=DELTAT/H
          TEMP=NSTEPS
          H=DELTAT/TEMP
C *****  OUTPUT SCALING
  230     DO 240 I=1,IDF
  240     ACCGS(I)=ACCISS(I)/386.
          ACCGS(2)=ACCISS(2)
C *****  ABSORBED POWER CALCULATION
          ABSPWR=0.
          IF(T.NE.0.)      ABSPWR=-100.*PWRVAR(1)/T
          CALL AVERAGE(ABSPWR,AVEPWR)
          IF(IFDRV.EQ.'N')         GO TO 260
          DISDRV=VAR(1)+DRVLEN*VAR(2)
          VELDRV=VAR(IDF+1)+DRVLEN*VAR(IDF+2)
C *****  RMS CALCULATION
  260     IF(IFRMS.EQ.'N')         GO TO 280
```

```
          DO 270 I=1,IDF
          SUMRMS(I)=SUMRMS(I)+ACCGS(I)**2
          RMS(I)=0.
  270     IF(T.NE.0.) RMS(I)=SQRT(SUMRMS(I)*DELTAT/T)
  280     IF(IFDRV.EQ.'N')          GO TO 300
          SDVRMS=SDVRMS+ACCDRV**2
          RMSDRV=0.
          IF(T.NE.0.)     RMSDRV=SQRT(SDVRMS*DELTAT/T)
C *****   PEAK ACCELERATION CALCULATION
  300     IF(IFPACC.EQ.'N')         GO TO 320
          DO 310 I=1,IDF
          ACCMAX(I)=AMAX1(ACCMAX(I),ACCGS(I))
  310     ACCMIN(I)=AMIN1(ACCMIN(I),ACCGS(I))
  320     IF(IFDRV.EQ.'N')          GO TO 340
          DRVMAX=AMAX1(DRVMAX,ACCDRV)
          DRVMIN=AMIN1(DRVMIN,ACCDRV)
C *****   PROFILE INPUT
  340     CALL FILIN(FID,JSTOP,SPACING,2)
C *****   PROGRAM OUTPUT
  350     IF(IFFILE.EQ.'Y') CALL FILWRT(FID,NPL,
         * FN2,AVEPWR)
          IF(T.LT.TPRINT)  GO TO 360
          CALL PRINT(FID,IFSTOP,OPTYPE,AVEPWR)
          TPRINT=TPRINT+TIP
          IF(IFSTOP.EQ.'Y')         GO TO 510
C *****   MAIN PROGRAM
  360     CALL SHIFT
C *****   SHIFT ADVANCES THE  Y  PROFILE ARRAY
          IF(JSTOP.EQ.2)  GO TO 500
          PROFIL(1)=YIN
          INDEX=-NSTEPS
          LDF=2*IDF
  370     DO 380 J=1,IDF
          K=J+IDF
  380     PY(J)=VAR(K)
          DO 390 I=1,NY
  390     Y(I)=PASTP(I)+((INDEX+NSTEPS+1)*
         *  (PROFIL(I)-PASTP(I)))/NSTEPS
          DO 440 I=1,4
          GO TO ISUB,(400,410,420,430)
  400     CALL WHEELS(FK,NYTEMP)
          GO TO 440
  410     CALL WHEELS(FK,NYTEMP)
          GO TO 440
  420     CALL M60(FK)
          GO TO 440
  430     CALL M113(FK)
  440     CALL RUNGE(P,Q,VAR,FK,LDF,I)
```

```
              DO 450 I=1,IDF
              K=I+IDF
       450    ACCISS(I)=(VAR(K)-PY(I))/H
              ACCDRV=(ACCISS(1)+DRVLEN*ACCISS(2))/386.
              IF(IFPWR.EQ.'N')        GO TO 480
              DO 470 I=1,4
              CALL POWER(PWRFK)
       470    CALL RUNGE(PP,QQ,PWRVAR,PWRFK,9,1)
       480    INDEX=INDEX+1
              IF(INDEX.NE.0)   GO TO 370
              T=T+DELTAT
              GO TO 230
       C ***** FINAL OUTPUT
       500    CALL PRINT(FID,IFSTOP,OPTYPE)
       510    IF(IFPACC.EQ.'Y')       CALL PEAKAC(NPL)
        995   FORMAT(A1)
        996   FORMAT(A5)
        997   FORMAT(I)
        998   FORMAT(F)
        999   FORMAT(/)
       9999   CALL EXIT
              END
```

```
      SUBROUTINE FILIN(FID,JS,SPACING,N)
C ***** THIS SUBROUTINE READS A NEW INPUT
C ***** VALUE (YIN), AND CHECKS FOR END OF FILE.
C
      DIMENSION FID(12)
      DIMENSION FYIN(10)
      DATA FYIN/10*0./
      IF(N.LT.2)  GO TO 130
  100 IY=IY+MM
  110 IF(IY.GT.10)  GO TO 120
      YIN=FYIN(IY)
      RETURN
  120 READ(22,900) (FYIN(I),I=1,10)
  900 FORMAT(10F)
      IF(EOFC) GO TO 9999
      IY=IY-10
      GO TO 110
  130 WRITE(6,901)
  901 FORMAT('+ DESIRED DELTAL: ',$)
      READ(5,998) DELTAL
      READ(22,902) SPACING,(FID(I),I=1,12),
     + (FYIN(J),J=1,10)
  902 FORMAT(1X,F/1X,12A5,/10F)
      MM=DELTAL/SPACING
      IY=0
      RETURN
 9999 JS=2
      RETURN
  998 FORMAT(F)
      END
```

```
      SUBROUTINE RUNGE(P,Q,X,FK,M,N)
C ***** THIS SUBROUTINE IS THE RKG ALGORITHM.
      DIMENSION P(1),Q(1),X(1),FK(1),A(4),B(4),C(4)
      DATA A/.5,.292893219,1.70710678,.166666667/
      DATA B/2.,1.,1.,2./
      DATA C/.5,.292893219,1.70710678,.5/
      TA=A(N)
      TB=B(N)
      TC=C(N)
      DO 100 I=1,M
      P(I)=TA*(FK(I)-TB*Q(I))
      X(I)=X(I)+P(I)
  100 Q(I)=Q(I)+3.*P(I)-TC*FK(I)
      RETURN
      END
```

```
      SUBROUTINE DATA(N)
C ***** THIS SUBROUTINE CONTAINS THE VEHICLE
C *****   PARAMETERS; IT ALSO CALLS GAMSUB
      DIMENSION ISETUP(5),DSIGMA(9,4)
      DIMENSION DVEHCL(2,4),DTHRSH(9,4),DGAMMA(9,4)
      DIMENSION DMASS(6,4),DVAR(9,4),DLEN(10,4)
      DIMENSION DEMASS(4),DINRTA(4),DDRVLN(4)
      DIMENSION DSLIM(4,3,2),DSSLO(5,3,2)
      DIMENSION DSINT(5,3,2)
      DIMENSION DDLIM(2,3,2),DDSLO(3,3,2)
      DIMENSION DDINT(3,3,2)
      DIMENSION DR(2)
      INTEGER DSETUP(5,4)
      REAL LEN,MASS,INRTIA
      EQUIVALENCE (ISETUP(1),NY)
      DATA DVEHCL/10HM-151 JEEP,10HM-35 TRUCK,
     1 10HM-60  TANK,10HM-113 TANK/
C ***** DSETUP'S ARE NY,IDF,NAXLES,NSEGS,IFHORZ
      DATA DSETUP/0,4,2,0,0,0,5,3,0,0,
     & 50,9,6,5,1,36,8,5,5,1/
      DATA DTHRSH/9*0.,9*0.,
     1 3.5,1.,0.,1.,3.5,4*0.,
     1 3.2,.9,0.,.9,3.2,4*0./
      DATA DGAMMA/9*0.,9*0.,
     1 3885.,4715.,5000.,4715.,3885.,4*0.,
     1 1500.,2000.,3500.,2000.,1500.,4*0./
      DATA DSIGMA/9*0.,9*0.,
     1 3145.,1670.,0.,-1670.,-3145.,4*0.,
     1 1500.,700.,0.,-700.,-1500.,4*0./
      DATA DLEN/44.3,40.7,8*0.,113.,39.,24.,24.,6*0.,
     1 77.,44.,11.,-22.,-55.,-88.,4*0.,
     1 52.,24.,0.,-28.,-65.,5*0./
      DATA DMASS/.27,.27,4*0.,
     1 1.191,2.08,2.05,3*0.,6*0.,6*0./
      DATA DEMASS/2.58,18.8,0.,0./
      DATA DINRTA/3282.,90876.,0.,0./
      DATA DDRVLN/0.,0.,25.,25./
      DATA DVAR/9*0.,9*0.,
     1 -5.79,-.0089,-.966,-.97,
     1 -.942,-.913,-.884,-.856,0.,
     1 -3.75,-.0087,-.76,-.78,-.76,-.73,-.68,2*0./
      DATA DSLIM/12*999.,-4.4,-3.65,3.65,4.4,
     1 -5.7,-5.1,5.1,5.7,-5.7,-5.1,5.1,5.7/
      DATA DSSLO/15*1500.,
     1 11771.43,3333.33,1145.2,3333.33,11771.43,
     1 46000.,9333.33,2509.8,9333.33,46000.,
     1 46000.,9333.33,2509.8,9333.33,46000./
```

```
                DATA DSINT/15*0.,
            1   66714.3,7986.65,0.0,-7986.65,-66714.3,
            1   243800.,34800.,0.0,-34800.,-243800.,
            1   243800.,34800.,0.0,-34800.,-243800./
                DATA DDLIM/4*999.,2*0.,
            1   -0.6,0.6,-0.6,0.6,-0.6,0.6/
                DATA DDSLO/9*42.,
            1   70.,1402.,40.,0.,1583.,0.,0.,1583.,0./
                DATA DDINT/9*0.,
            1   -800.,0.,820.,-950.,0.,950.,-950.,0.,950./
                DATA DR/15.,19./
                DO 100 I=1,2
      100       VEHQID(I)=DVEHCL(I,N)
                DO 110 I=1,5
      110       ISETUP(I)=DSETUP(I,N)
                DO 120 I=1,IDF
                LEN(I)=DLEN(I,N)
      120       VAR(I)=DVAR(I,N)
                DO 130 I=1,NAXLES
      130       MASS(I)=DMASS(I,N)
                FMASS=DFMASS(N)
                INRTIA=DINRTA(N)
                DRVLEN=DDRVLN(N)
                IF(N.LE.2)  GO TO 150
                DO 140 I=1,NSEGS
                THRESH(I)=DTHRSH(I,N)
                SIGMA(I)=DSIGMA(I,N)
      140       GAMMA(I)=DGAMMA(I,N)
                RETURN
      150       DO 200 I=1,NAXLES
                DO 160 J=1,4
      160       SLIMIT(J,I)=DSLIM(J,I,N)
                DO 170 J=1,5
                SSLOPE(J,I)=DSSLO(J,I,N)
      170       SINT(J,I)=DSINT(J,I,N)
                DO 180 J=1,2
      180       DLIMIT(J,I)=DDLIM(J,I,N)
                DO 190 J=1,3
                DSLOPE(J,I)=DDSLO(J,I,N)
      190       DINT(J,I)=DDINT(J,I,N)
      200       CONTINUE
                R=DR(N)
                CALL GAMSUB(N,R)
                DO 210 I=1,NAXLES
                VAR(I+2)=-YY(I)
                WEIGHT(I)=WEIGHT(I)-MASS(I)*386.
                IF(NAXLES.LT.3) GO TO 210
                IF(I.LT.2)  GO TO 210
                WEIGHT(I)=2.*WEIGHT(I)
      210       SPDEF(I)=-WEIGHT(I)/SSLOPE(3,I)
```

```
            WB=LEN(1)+LEN(2)
            T1=VAR(3)+SPDEF(1)
            T2=VAR(4)+SPDEF(2)
            IF(N.LT.2)  GO TO 220
            T3=VAR(5)+SPDEF(3)
            T2=(T2*LEN(4)+T3*LEN(3))/(LEN(3)+LEN(4))
 220        VAR(1)=(T1*LEN(2)+T2*LEN(1))/WB
            VAR(2)=(T2-T1)/WB
            WRITE(6,800) (VAR(I),I=1,IDF)
     800    FORMAT(5F10.5)
            RETURN
            END
```

```
                 SUBROUTINE GAMSUB(N,R)
                 DIMENSION VEHWT(2),RF(2),WDF(2),WDR(2,2)
                 DIMENSION D(0/30),DPRIME(0/30),THR(0/30)
                 DIMENSION DELTA(0/90)
                 REAL LEN,MASS
                 DO 100 I=0,30,1
                 D(I)=I*DELTAL
      100        IF((D(I)/R).GT.0.788)   GO TO 110
      110        NSEGS=I*2-1
                 NY=(LEN(1)+LEN(2)+LEN(4))/DELTAL+NSEGS
                 KK=I-1
                 KZERO=I
                 Q=LEN(1)+LEN(2)
                 WEIGHT(1)=(FMASS*(LEN(2)/Q)+MASS(1))*386.
                 WEIGHT(2)=(((FMASS*(LEN(1)/Q))/(NAXLES-1))
               & +MASS(2))*386.
                 IF(NAXLES.LT.3)  GO TO 120
                 WEIGHT(2)=WEIGHT(2)/2.
                 WEIGHT(3)=(((FMASS*(LEN(1)/Q))/(NAXLES-1))
               & +MASS(3))*386./2.
      120        WRITE(6,900) VEHOID
           900   FORMAT(' FROM FORCE-DEFL CURVES FOR '/1X,
               * 2A5,' TIRES, ENTER DEFLECTIONS: '/)
                 DO 135 I=1,NAXLES
                 WRITE(6,901) WEIGHT(I)
           901   FORMAT('+  LOAD = ',F8.3,3X,$)
      135        READ(5,902) YY(I)
           902   FORMAT(F)
                 SUM5=0.
                 SUM6=0.
                 SUM7=0.
                 DO 130 I=0,KK,1
                 DPRIME(I)=SQRT(R**2.-D(I)**2.)
                 COSTHETA=DPRIME(I)/R
                 THR(I)=R-DPRIME(I)
                 DELTA(I)=YY(1)-THR(I)
                 IF(DELTA(I).LT.0.0)  DELTA(I)=0.0
                 SUM5=SUM5+DELTA(I)*COSTHETA
                 J=I+KK+1
                 DELTA(J)=YY(2)-THR(I)
                 IF(DELTA(J).LT.0.0)      DELTA(J)=0.0
                 SUM6=SUM6+DELTA(J)*COSTHETA
                 JJ=J+KK+1
                 DELTA(JJ)=YY(3)-THR(I)
                 IF(DELTA(JJ).LT.0.0)   DELTA(JJ)=0.0
                 SUM7=SUM7+DELTA(JJ)*COSTHETA
      130        CONTINUE
      C$$$$$
```

```
      SUM5=SUM5*2.-DELTA(0)
      SUM6=SUM6*2.-DELTA(KK+1)
      SUM7=SUM7*2.-DELTA(2*KK+2)
      SPKF=WEIGHT(1)/SUM5
      SPKR1=WEIGHT(2)/SUM6
      SPKR2=0.0
      IF(N.EQ.1)  GO TO 136
      SPKR1=2*SPKR1
      SPKR2=2*WEIGHT(3)/SUM7
C$$$$$
  136 DO 140 I=KK,0,-1
      THRESH(KZERO-I)=THR(I)
  140 THRESH(KZERO+I)=THR(I)
      DO 150 I=1,NSEGS
      THRESH(I+NSEGS)=THRESH(I)
      J=I+2*NSEGS
  150 THRESH(J)=THRESH(I)
C$$$$$
      DO 160 I=KK,0,-1
      J=KZERO-I
      JJ=KZERO+I
      COSTHETA=DPRIME(I)/R
      GAMMA(J)=SPKF*COSTHETA
      GAMMA(JJ)=GAMMA(J)
      J1=J+NSEGS
      JJ1=JJ+NSEGS
      GAMMA(J1)=SPKR1*COSTHETA
      GAMMA(JJ1)=GAMMA(J1)
      J2=J1+NSEGS
      JJ2=JJ1+NSEGS
      GAMMA(J2)=SPKR2*COSTHETA
  160 GAMMA(JJ2)=GAMMA(J2)
C$$$$$
      WRITE(21,930) ((I,D(I)),I=0,KK)
  930 FORMAT(50(' D(',I2,') = 'F8.2/)/)
      WRITE(21,931)
  931 FORMAT(//'  I            THRESH(I)'
     +  '        GAMMA(I)'/6X,'FRONT:')
      DO 240 I=1,2*NSEGS
      WRITE(21,932) I,THRESH(I),GAMMA(I)
  932 FORMAT(1X,I2,1X,2F15.3)
  240 IF(I.EQ.NSEGS)  WRITE(21,933)
  933 FORMAT(/6X,'REAR1')
      IF(N.EQ.1)  GO TO 9999
      WRITE(21,934)
  934 FORMAT(/6X'REAR2:')
      DO 260 I=2*NSEGS+1,3*NSEGS,1
  260 WRITE(21,932) I,THRESH(I),GAMMA(I)
 9999 RETURN
      END
```

```
      SUBROUTINE PEAKAC(NPL)
C *****THIS SUBROUTINE WRITES THE PEAK
C *****ACCELERATION VALUES
      WRITE(6,901)
  901 FORMAT(' PEAK ACCELERATION VALUES'/9X,
     +'MAXIMUM  MINUMUM')
      WRITE(6,902) (ACCMAX(I),ACCMIN(I),I=1,2)
  902 FORMAT(' C-G   '2F9.4,/' PITCH ',2F9.4)
      DO 110 I=1,NAXLES
      J=I+2
  110 WRITE(6,903) I,ACCMAX(J),ACCMIN(J)
  903 FORMAT(' AXLE',I1,1X,2F9.4)
      IF(IFHORZ.EQ.1)  WRITE(6,904) ACCMAX(IDF),
     +ACCMIN(IDF)
  904 FORMAT(' HORIZ ',2F9.4)
      IF(IFDRV.EQ.'Y') WRITE(6,905) DRVMAX,DRVMIN
  905 FORMAT(' DRIVER',2F9.4)
      IF(IFFILE.EQ.'N')      RETURN
      WRITE(21,901)
      WRITE(21,902) (ACCMAX(I),ACCMIN(I),I=1,2)
      DO 120 I=1,NAXLES
      J=I+2
  120 WRITE(21,903) I,ACCMAX(J),ACCMIN(J)
      IF(IFHORZ.EQ.1) WRITE(21,904) ACCMAX(IDF),
     +ACCMIN(IDF)
      IF(IFDRV.EQ.'Y') WRITE(21,905) DRVMAX,DRVMIN
      RETURN
      END
```

```
      SUBROUTINE AVERAGE(X,Y)
C *****THIS SUBROUTINE COMPUTES THE
C *****AVERAGE OF ANY INPUT X
      DATA N/0/
      DATA SUM/0./
      N=N+1
      SUM=SUM+X
      Y=SUM/N
      RETURN
      END
```

```
      SUBROUTINE PRINT(FID,IFSTOP,OPTYPE,AP)
C ***** THIS SUBROUTINE HANDLES THE TTY PRINTOUT
      DIMENSION FID(12),HEAD(5),VID(3)
      DIMENSION VOUT(4)
      DIMENSION DRIVER(4)
      EQUIVALENCE (DRIVER(1),DISDRV)
      DATA VID/15HABSORBED POWER=/
      DATA HEAD/'DISPL','VELOC','ACCEL',
     + 'RMSAC','     '/
      DATA IFIRST/0/
      IF(OPTYPE.EQ.'Y')  GO TO 170
      IF(IFIRST.EQ.1) GO TO 110
      K=4
      IF(IFRMS.EQ.'N')          K=3
      WRITE(6,901) VELMPH,VELIPS,DELTAL,DELTAT,
     + NSTEPS,H
  901 FORMAT(///' VELOCITY='F5.2,' MPH ('
     + F6.2,' IPS)'/, ' DELTA-L=',F5.3,3X,
     + 'DELTA-T=',F6.4/' NSTEPS=',I4,4X,
     + 'H=',F7.6)
      WRITE(6,902) VEHQID,FID
  902 FORMAT(' VEHICLE IS: '2A5/' INPUT PROFILE IS
     : '/1X,12A5)
      WRITE(6,903) (HEAD(I),I=1,K)
  903 FORMAT(//7X,4(5X,A5))
      IFIRST=1
  110 IF(IFPWR.EQ.'N')          GO TO 120
      WRITE(6,904) T,PROFIL(1),VID,ABSPWR
  904 FORMAT(/' TIME='F6.3,
     + '  INPUT='F7.3,3X,3A5,F7.3)
      GO TO 130
  120 WRITE(6,904) T,PROFIL(1)
  130 CALL VARFIX(1,VOUT)
      WRITE(6,905) (VOUT(I),I=1,K)
  905 FORMAT(/' C-G   ',4F10.5)
      CALL VARFIX(2,VOUT)
      WRITE(6,906) (VOUT(I),I=1,K)
  906 FORMAT(' PITCH ',4F10.5)
      DO 140 L=1,NAXLES
      N=2+L
      CALL VARFIX(N,VOUT)
  140 WRITE(6,907) L,(VOUT(I),I=1,K)
  907 FORMAT(' AXLE',I1,1X,4F10.5)
      IF(IFHORZ.EQ.0) GO TO 150
      N=N+1
      CALL VARFIX(N,VOUT)
      WRITE(6,908),(VOUT(I),I=1,K)
  908 FORMAT(' HORIZ ',4F10.5)
```

```
150     IF(IFDRV.EQ.'N')        GO TO 160
        WRITE(6,909) (DRIVER(I),I=1,K)
  909   FORMAT(' DRIVER',4F10.5)
160     WRITE(6,910)
  910   FORMAT(' STOP? ',$)
        ACCEPT 996, IFSTOP
  996   FORMAT(A1)
        KK=0
        RETURN
170     IF(IFIRST.EQ.0)  WRITE(6,911)
  911   FORMAT(//'  TIME ABSPWR AVEPWR')
        IFIRST=1
        WRITE(6,912)  T,ABSPWR,AVEPWR
  912   FORMAT(1X,F5.2,2F7.2)
        KK=KK+1
        IF(KK.GE.10)  GO TO 160
        END
```

```
        SUBROUTINE VARFIX(I,VOUT)
C *****  THIS SUBROUTINE IS CALLED BY PRINT
C *****     TO SELECT THE VARIABLES TO BE PRINTED.
        DIMENSION VOUT(4)
        VOUT(1)=VAR(I)
        N=I+IDF
        VOUT(2)=VAR(N)
        VOUT(3)=ACCGS(I)
        VOUT(4)=RMS(I)
        RETURN
        END
```

```
      SUBROUTINE POWER(FK)
C ***** THIS SUBROUTINE CALCULATES ABSORBED POWER.
      DIMENSION FK(9)
      U2=-67.743*ACCDRV-1.042*PWRVAR(8)
      U1=-U2-3.246*PWRVAR(6)
      U0=-U1+1.318*PWRVAR(4)
      FK(1)=H*(.00873*PWRVAR(2)*PWRVAR(3))
      FK(2)=H*(-4.99484*ACCDRV)
      FK(3)=H*(-100.*U0-59.*PWRVAR(3))
      FK(4)=H*(-13.*U1+71.6*PWRVAR(5)
     + -53.49*PWRVAR(4))
      FK(5)=H*(-100.*U1-47.78*U0)
      FK(6)=H*(-10.*U2-78.59*PWRVAR(7)
     + -55.28*PWRVAR(6))
      FK(7)=H*(-10.*U2-6.259*U1)
      FK(8)=H*(-677.43*ACCDRV-388.8*PWRVAR(9)
     + -46.67*PWRVAR(8))
      FK(9)=H*(-67.743*ACCDRV-2.742*U2)
      RETURN
      END
```

```
      SUBROUTINE SHIFT
C ***** THIS SUBROUTINE ADVANCES THE PROFILE
C *****    UNDER THE VEHICLE
      DO 100 I=NY,2,-1
      PASTP(I-1)=PROFIL(I-1)
  100 PROFIL(I)=PROFIL(I-1)
      RETURN
      END
```

```
      SUBROUTINE FILWRT(FID,NPL,FN2,AP)
C ***** THIS SUBROUTINE HANDLES THE OUTPUT
C *****    TO AN EXTERNAL FILE
      DIMENSION HEAD1(6),HEAD2(2)
      DIMENSION VOUT(10),FID(12)
      DATA IFIRST/0/
      DATA HEAD1/'AXLE1','AXLE2','AXLE3','AXLE4',
     + 'AXLE5','AXLE6'/
      DATA HEAD2/'H,C-G','V,DRV'/
      IF(IFIRST.NE.0) GO TO 130
      WRITE(21,902) VELMPH,VELIPS,DELTAL,DELTAT,
     + NSTEPS,H
  902 FORMAT(/,' VELOCITY = ',F6.2,
     + ' MPH ('F6.2,' IPS) ',7X,'DELTA-L='
     + ,F5.3,' INCHES,',9X,'DELTA-T=',F10.8,
     + ' SECONDS',//' RUNGE-KUTTA-GILL INTEGRATION:
     + NUMBER OF STEPS=',I4,10X,'STEP SIZE (H)=',
     + E12.6//)
      WRITE(21,903) VEHOID,FID
  903 FORMAT(//' VEHICLE IS: ',2A5,//
     + ' PROFILE IS: ',12A5)
      IFIRST=1
      KK=2
      IF(IFDRV.NE.'N')        GO TO 120
      HEAD2(2)=0.
      KK=KK-1
  120 IF(IFHORZ.NE.0.)        GO TO 160
      HEAD2(1)=HEAD2(2)
      KK=KK-1
      GO TO 160
  130 IF(NPL.LT.50)    GO TO 170
  140 IF(NPL.GE.54)    GO TO 150
      WRITE(21,904)
  904 FORMAT(1H )
      NPL=NPL+1
      GO TO 140
  150 NPL=0
  160 WRITE(21,905) (HEAD1(I),I=1,NAXLES),
     + (HEAD2(I),I=1,KK)
  905 FORMAT(1H1,'  TIME   Y(I)',14X,
     + 'V,C-G     PITCH',4X,8(A5,4X))
      WRITE(21,904)
      NPL=NPL+2
  170 DO 180 I=1,IDF
  180 VOUT(I)=VAR(I)
      J=IDF
      IF(IFDRV.EQ.'N')        GO TO 190
      J=J+1
      VOUT(J)=DISDRV
```

```
  190   WRITE(21,906) T,PROFIL(1),(VOUT(I),I=1,J)
    906 FORMAT(/1X,F7.4,F6.2,'  DISPL    ',10F9.4)
        DO 200 I=1,IDF
        K=I+IDF
  200   VOUT(I)=VAR(K)
        IF(IFDRV.EQ.'N')        GO TO 210
        VOUT(J)=VELDRV
  210   WRITE(21,907) AP,(VOUT(I),I=1,J)
    907 FORMAT(' AVEPWR=',F6.2,'  VELOCITY',10F9.4)
        DO 220 I=1,IDF
  220   VOUT(I)=ACCGS(I)
        IF(IFDRV.EQ.'N')        GO TO 230
        VOUT(J)=ACCDRV
  230   IF(IFPWR.NE.'N')        GO TO 240
        WRITE(21,908) (VOUT(I),I=1,J)
    908 FORMAT(10X,'ACCELERATION',2X,10F9.4)
        GO TO 250
  240   WRITE(21,909) ABSPWR,(VOUT(I),I=1,J)
    909 FORMAT(' POWER=',F6.2,'   ACCEL    ',10F9.4)
  250   NPL=NPL+4
        IF(IFRMS.EQ.'N')        RETURN
        DO 260 I=1,IDF
  260   VOUT(I)=RMS(I)
        IF(IFDRV.EQ.'N')        GO TO 270
        VOUT(J)=RMSDRV
  270   WRITE(21,910) (VOUT(I),I=1,J)
    910 FORMAT(16X,'RMS ACC ',10F9.4)
        NPL=NPL+1
        RETURN
    996 FORMAT(A5)
        END
```

```
             SUBROUTINE WHEELS(FK,NYTEMP)
C *****      THIS SUBROUTINE IS THE GENERALIZED
C *****      WHEELED VEHICLE MODEL
             DIMENSION TEMP(3),FK(18),NYTEMP(3)
             REAL LEN,MASS,INRTIA
             DATA IFIRST/0/
C *****      RESULTING AXLE FORCES
99           DO 110 I=1,NAXLES
             FORCW(I)=0.
             DO 100 J=1,NSEGS
             JJ=J+NYTEMP(I)
             JJJ=J+(I-1)*NSEGS
             TEMP0=Y(JJ)-VAR(2+I)-THRESH(JJJ)
             IF(TEMP0.LT.0.) TEMP0=0.0
100          FORCW(I)=FORCW(I)+GAMMA(JJJ)*TEMP0
110          CONTINUE
C *****
             UU=LEN(3)+LEN(4)
             IF(UU.LE.0.) GO TO 120
             U=(VAR(4)-VAR(5))/UU
             BETA=ATAN(U)
             DBETA=(VAR(4+IDF)-VAR(5+IDF))/(UU*(1.+U*U))
             TEMP2=SIN(BETA)
             TEMP4=COS(BETA)
120          TEMP1=SIN(VAR(2))
             TEMP3=COS(VAR(2))
C *****      SUSPENSION SPRING DEFLECTION  (SPDEF)
             SPDEF(1)=VAR(3)-VAR(1)-LEN(1)*TEMP1
             SPDEF(2)=VAR(4)-VAR(1)+LEN(2)*TEMP1
          &  -LEN(3)*TEMP2
             IF(NAXLES.LT.3) GO TO 130
             SPDEF(3)=VAR(5)-VAR(1)+LEN(2)*TEMP1
          &  +LEN(4)*TEMP2
C *****      SUSPENSION SPRING RATE OF DEFLECTION  (DSPDF
130          DSPDF(1)=VAR(3+IDF)-VAR(1+IDF)
          &  -LEN(1)*VAR(2+IDF)*TEMP3
             DSPDF(2)=VAR(4+IDF)-VAR(1+IDF)
          &  +LEN(2)*VAR(2+IDF)*TEMP3-LEN(3)*DBETA*TEMP4
             IF(NAXLES.LT.3)  GO TO 140
             DSPDF(3)=VAR(5+IDF)-VAR(1+IDF)
          &  +LEN(2)*VAR(2+IDF)*TEMP3+LEN(4)*DBETA*TEMP4
C *****      SUSPENSION SPRING FORCES
140          DO 160 I=1,NAXLES
             K=5
             DO 150 M=1,4
             IF(SPDEF(I).LT.SLIMIT(M,I))  GO TO 155
150          CONTINUE
             GO TO 160
```

```
      155       K=M
      160       FORCK(I)=SSLOPE(K,I)*SPDEF(I)+SINT(K,I)
      C *****   SUSPENSION DAMPING FORCES
                DO 180 I=1,NAXLES
                K=3
                DO 170 M=1,2
                IF(DSPDF(I).LT.DLIMIT(M,I))  GO TO 175
      170       CONTINUE
                GO TO 180
      175       K=M
      180       DAMP(I)=DSLOPE(K,I)*DSPDF(I)+DINT(K,I)
      C *****   DIFFERENTIAL EQUATIONS
      C *****     FK(1) AND FK(1+IDF)---CG MOTION
      C *****     FK(2) AND FK(2+IDF)---PITCH MOTION
      C *****     FK(3) AND FK(3+IDF)---AXLE1 MOTION
      C *****     .      .      .      .      .      .
      C *****     FK(N) AND FK(N+IDF)---AXLEN MOTION
                DO 190 I=1,IDF
      190       FK(I)=H*VAR(I+IDF)
                STEMP=0.
                DO 200 I=1,NAXLES
                TEMP(I)=FORCK(I)+DAMP(I)
                STEMP=STEMP+TEMP(I)
      200       FK(I+2+IDF)=H*(FORCH(I)-TEMP(I)
              & -MASS(I)*386.)/MASS(I)
                FK(1+IDF)=H*(STEMP-FMASS*386.)/FMASS
                FK(2+IDF)=H*(LEN(1)*TEMP(1)
              & -LEN(2)*TEMP(2))/INRTIA
                RETURN
                END



      C$$$$$



                SUBROUTINE M113(FK)
      C *****   THIS SUBROUTINE IS FOR A TRACKED
      C *****     VEHICLE, NOT USED HERE
                END
      C
      C
      C
      C
      C
                SUBROUTINE M60(FK)
      C *****   THIS SUBROUTINE IS FOR A TRACKED
      C *****      VEHICLE, NOT USED HERE
                END
```

APPENDIX C:

The vehicle parameters necessary for wheeled vehicle simulation are listed in the same order as in subroutine data. The variable names are written in all capital letters. See Figures C1 through C3 for clarification. Those parameters in DATA, not listed herein, are not necessary for wheeled vehicle simulation. This appendix is intended as a work sheet for use in obtaining data for simulation of a new vehicle.

A. DSETUP

1. NY - number of profile points under vehicle (computed) 0.0

2. IDF - degrees of freedom . . . . ____

   = 4 for 2-axle vehicles

   = 5 for 3-axle vehicles

3. NAXLES - number of axles . . . . ____

4. NSEGS - number of segments in tire model (computed) 0.0

5. IFHORZ 0.0

B. DLEN

1. LEN(1) - ℓ_1: horizontal distance from front axle to center of gravity . . . . . ____ inches

2. LEN(2) - ℓ_2: horizontal distance from CG to rear axle (or center of rear bogie assembly) . . . . ____ inches

3. LEN(3) - ℓ_2: horizontal distance from center of rear bogie assembly to center of 2nd axle (= 0 for 2 axles) . . . . ____ inches

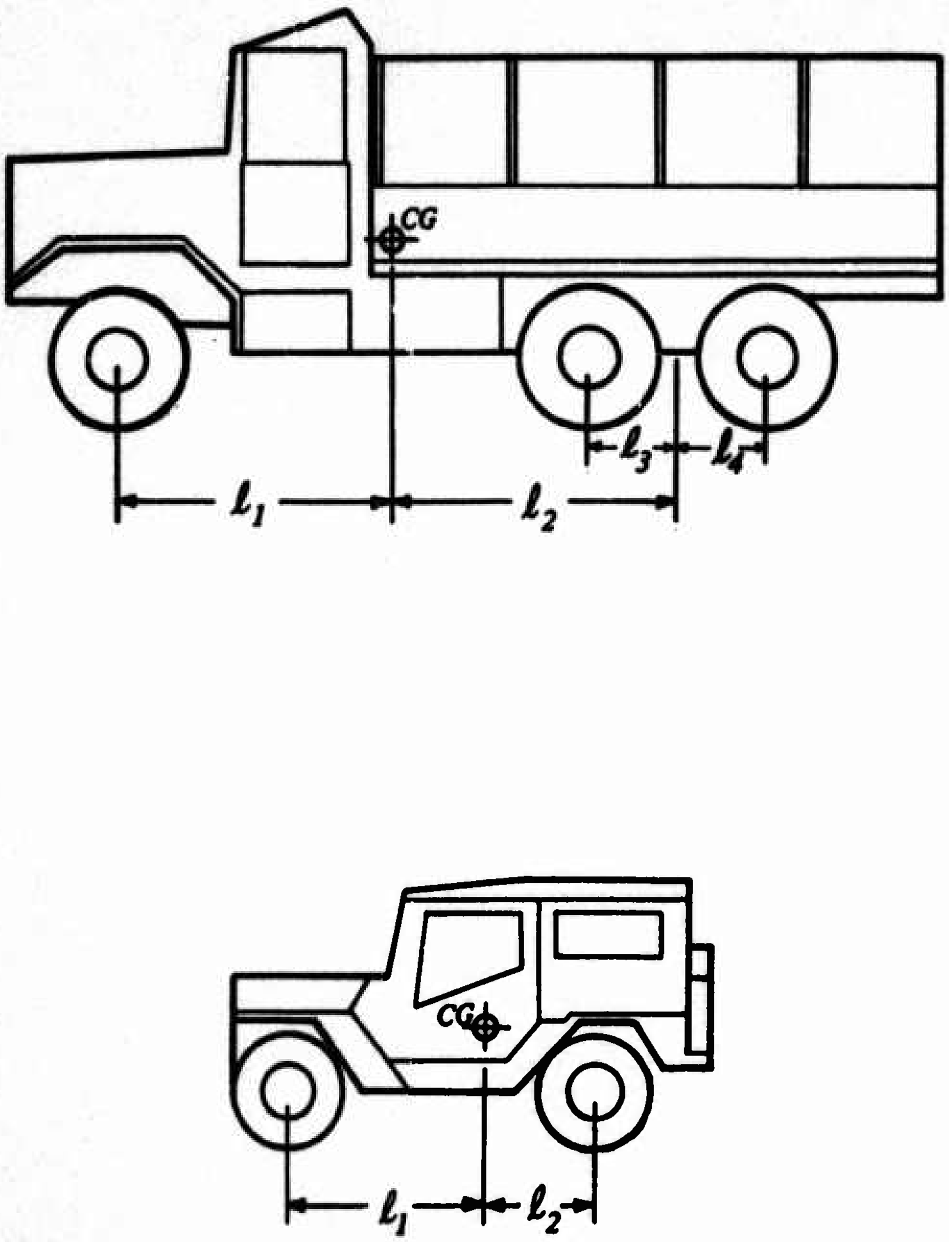


Figure C1. Geometrical Vehicle Parameters

4. LEN(4) - ℓ: horizontal distance from center of rear bogie to center of 3rd axle (= 0 for 2-axles) . . . . ____ inches

5. LEN(5) thru LEN(9) . . . . 0.0 inches

C. DMASS - Unsprung masses

1. MASS(1) - M_1: Mass of front axle including wheels . . . . ____ $\frac{\text{lb-sec}^2}{\text{inches}}$

2. MASS(2) - M_2: Mass of second axle . . . . . ____ $\frac{\text{lb-sec}^2}{\text{inches}}$

3. MASS(3) - M_3: Mass of third axle . . . . . ____ $\frac{\text{lb-sec}^2}{\text{inches}}$

4. MASS(4) - MASS(6) . . . . ____ $\frac{\text{lb-sec}^2}{\text{inches}}$

D. DFMASS-M_o: Sprung mass . . . . ____ $\frac{\text{lb-sec}^2}{\text{inches}}$

E. DINRTA-I_o: Pitch moment of inertia of spring mass about CG . . . . ____ $\frac{\text{lb-sec}^2}{\text{inches}}$

F. DESCRIPTION OF SUSPENSION SPRING FORCE FUNCTION (see Figure C2)

1. DSLIM - limits of regions

a. Front axle:

(1) SLIMIT(1,1) - x_{11} . . . . ____ inches

(2) SLIMIT(2,1) - x_{21} . . . . ____ inches

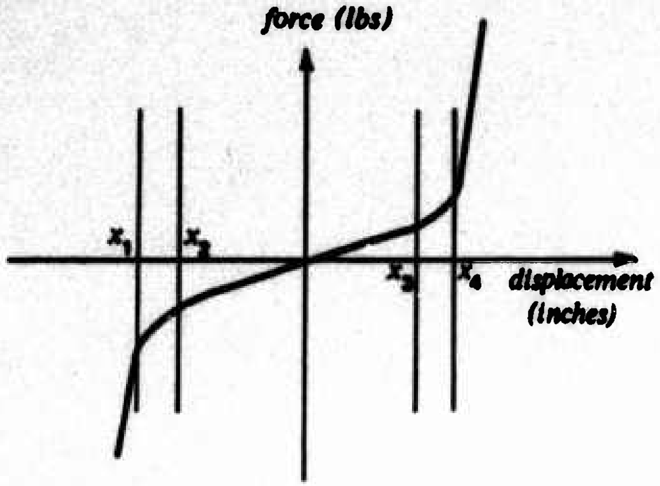
(3) SLIMIT(3,1) - x_{31} . . . . ____ inches

(4) SLIMIT(4,1) - x_{41} . . . . ____ inches

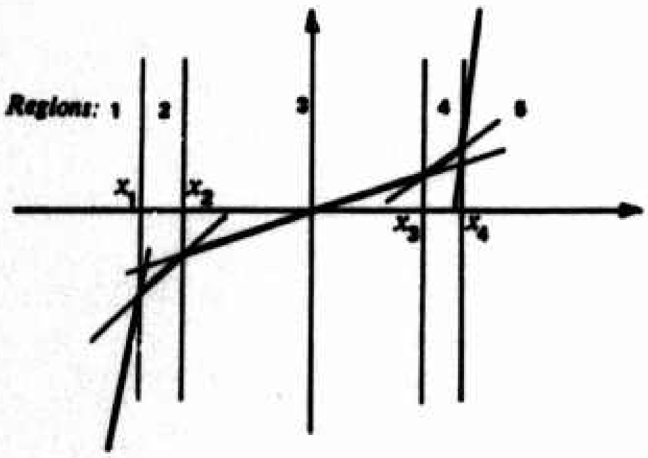
b. Second axle:

(1) SLIMIT(1,2) - x_{12} . . . . ____ inches

(2) SLIMIT(2,2) - x_{22} . . . . ____ inches



Typical curve



Segmented linear approximation

Equations of lines are:

$$F_{ij} = m_{ij}\delta_j + C_{ij}$$

i = region number (=1,2,3,4,5)

j = axle number (1,2,3)

Figure C2. Segmented-Linear Spring Force Approximation

(3) SLIMIT(3,2) - x_{32} . . . . ____ inches

(4) SLIMIT(3,2) - x_{42} . . . . ____ inches

c. Third axle (if 2-axles, these are zeroes)

(1) SLIMIT(1,3) - x_{13} . . . . ____ inches

(2) SLIMIT(2,3) - x_{23} . . . . ____ inches

(3) SLIMIT(3,3) - x_{33} . . . . ____ inches

(4) SLIMIT(4,3) - x_{43} . . . . ____ inches

2. DSSLO - Slope of lines which approximate the force vs. deflection curve of the suspension springs.

a. Front axle:

(1) SSLOPE(1,1) - m_{11} . . . . ____

(2) SSLOPE(2,1) - m_{21} . . . . ____

(3) SSLOPE(3,1) - m_{31} . . . . ____

(4) SSLOPE(4,1) - m_{41} . . . . ____

(5) SSLOPE(5,1) - m_{51} . . . . ____

b. Second axle:

(1) SSLOPE(1,2) - m_{12} . . . . ____

(2) SSLOPE(2,2) - m_{22} . . . . ____

(3) SSLOPE(3,2) - m_{32} . . . . ____

(4) SSLOPE(4,2) - m_{42} . . . . ____

(5) SSLOPE(5,2) - m_{52} . . . . ____

c. Third axle:

(1) SSLOPE(1,3) - m_{13} . . . . ____

(2) SSLOPE(2,3) - m_{23} . . . . ____

(3) SSLOPE(3,3) - m_{33} . . . . ____

(4) SSLOPE(4,3) - m_{43} . . . . ____

(5) SSLOPE(5,3) - m_{53} . . . . ____

3. DSINT - Intercepts of force vs deflection lines.

a. Front axle:

(1) SINT(1,1) - c_{11} . . . . ____

(2) SINT(2,1) - c_{21} . . . . ____

(3) SINT(3,1) - c_{31} . . . . ____

(4) SINT(4,1) - c_{41} . . . . ____

(5) SINT(5,1) - c_{51} . . . . ____

b. Second axle:

(1) SINT(1,2) - c_{12} . . . . ____

(2) SINT(2,2) - c_{22} . . . . ____

(3) SINT(3,2) - c_{32} . . . . ____

(4) SINT(4,2) - c_{42} . . . . ____

(5) SINT(5,2) - c_{52} . . . . ____

c. Third axle:

(1) SINT(1,3) - c_{13} . . . . ____

(2) SINT(2,3) - c_{23} . . . . ____

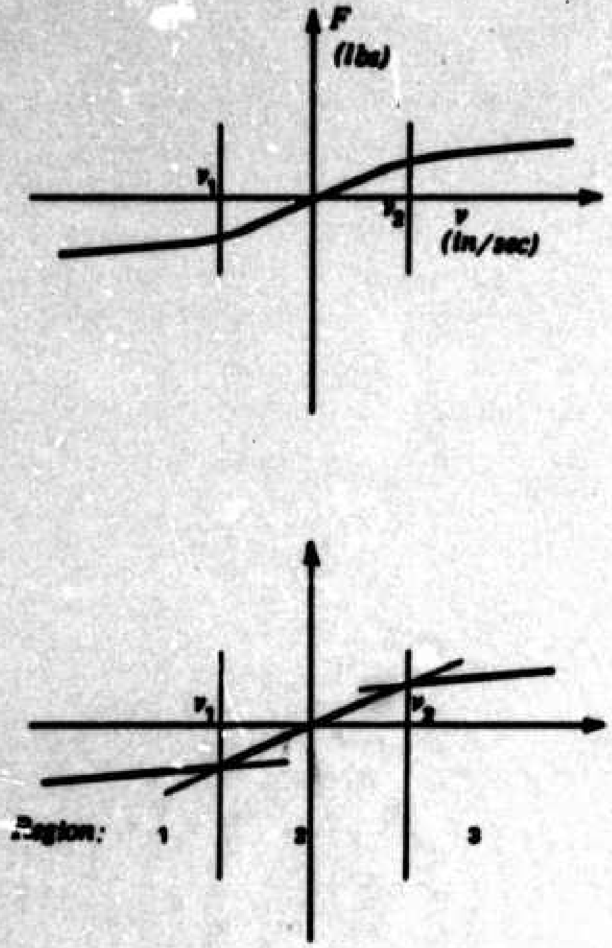
(3) SINT(3,3) - c_{33} . . . . ____

(4) SINT(4,3) - c_{43} . . . . ____

(5) SINT(5,3) - c_{53} . . . . ____

G. DESCRIPTION OF SUSPENSION DAMPING FORCE FUNCTION (see Figure C3)

1. DDLIM - Limits of regions.



Typical curve

Segmented linear approximation

Equations of lines are:

$$F_{ij} = n_{ij}\dot{\delta}_j + d_{ij}$$

i = region number (=1,2,3)

j = axle number (=1,2,3)

Figure C3. Segmented-Linear Damping Force Approximation

a. Front axle:

(1) DLIMIT(1,1) - v_{11} . . . . ____ in/sec

(2) DLIMIT(2,1) - v_{21} . . . . ____ in/sec

b. Second axle:

(1) DLIMIT(1,2) - v_{12} . . . . ____ in/sec

(2) DLIMIT(2,2) - v_{22} . . . . ____ in/sec

c. Third axle:

(1) DLIMIT(1,3) - v_{13} . . . . ____ in/sec

(2) DLIMIT(2,3) - v_{23} . . . . ____ in/sec

2. DDSLO - Slope of lines which approximate the force-velocity curve.

a. Front axle:

(1) DSLOPE(1,1) - n_{11} . . . . ____

(2) DSLOPE(2,1) - n_{21} . . . . ____

(3) DSLOPE(3,1) - n_{31} . . . . ____

b. Second axle:

(1) DSLOPE(1,2) - n_{12} . . . . ____

(2) DSLOPE(2,2) - n_{22} . . . . ____

(3) DSLOPE(3,2) - n_{32} . . . . ____

c. Third axle:

(1) DSLOPE(1,3) - n_{13} . . . . ____

(2) DSLOPE(2,3) - n_{23} . . . . ____

(3) DSLOPE(3,3) - n_{33} . . . . ____

3. DDINT - Intercepts of lines on damping force vs. deflection velocity curves.

a. Front axle:

(1) DINT(1,1) - d_{11} . . . . ______

(2) DINT(2,1) - d_{21} . . . . ______

(3) DINT(3.1) - d_{31} . . . . ______

b. Second axle:

(1) DINT(1,2) - d_{12} . . . . ______

(2) DINT(2,2) - d_{22} . . . . ______

(3) DINT(3,2) - d_{32} . . . . ______

c. Third axle:

(1) DINT(1,3) - d_{13} . . . . ______

(2) DINT(2,3) - d_{23} . . . . ______

(3) DINT(3,3) - d_{33} . . . . ______

H. DR - The undeflected tire radius in inches . . . ______ inches

APPENDIX D (Documentation of PWRPLT)

I. DESCRIPTION: This program plots the absorbed power and the average absorbed power against time. It stores the plot in a file on disc which can be printed on a line-printer with 130 characters per line.

II. INPUTS AND OPERATING INSTRUCTIONS:

A. Prior to execution: none

B. During execution:

1. IFILE: The name of the file which contains the power and absorbed power. This file <u>must</u> be in the format of the detailed output file from VEH, with options absorbed power and RMS accelerations specified. This file name must be 5 or less characters long.

2. OFILE: Desired name of plot file; not over 5 characters.

III. OUTPUTS: One file containing a plot of absorbed power and average power vs. time.

IV. SAMPLE EXECUTION:

```
.EX PWRPLT
FORTRAN: PWRPLT
LOADING

LOADER 4K CORE
EXECUTION

IFILE: FILEX

OFILE: PWRX
END OF FILE ON DSK1
(//F,/8X,F,/7X,F)
LAST FORTRAN I-O AT USER LOC 000267

EXECUTION TIME:          8.20 SEC.
TOTAL ELAPSED TIME:     45.20 SEC.
NO EXECUTION ERRORS DETECTED

EXIT
```

V. PROGRAM LISTING:

```
           DIMENSION X(120),T(10)
           DIMENSION P(10),AP(10)
           DATA X/120*' '/
           STAR='*'
           PLUS='+'
           BLANK=' '
           WRITE(6,900)
   900     FORMAT(' IFILE: ',S)
           ACCEPT 995, FN1
   995     FORMAT(A5)
           CALL IFILE(21,FN1)
           WRITE(6,905)
   905     FORMAT(' OFILE: ',S)
           ACCEPT 995, FN2
           CALL OFILE(22,FN2)
C
           WRITE(22,940) FN1
   940     FORMAT(' GRAPH OF TIME VS ABSORBED'
         + ' POWER FOR ',A5,'.DAT'//)
           WRITE(22,945)  (I,I=0,22)
   945     FORMAT('*','TIME ',I1,22(3X,I2),
         + 'POWER  AVE'/)
 105       DO 110 J=1,300
           READ(21,995) TEST
           IF(EOFC) GO TO 9999
 110       IF(TEST.EQ.'1  TI')  GO TO 120
           GO TO 9999
 120       DO 130 I=1,10
           READ(21,920) T(I),AP(I),P(I)
           IF(EOFC)  GO TO 135
 130       CONTINUE
   920     FORMAT(//F,/8X,F,/7X,F)
           JJ=10
           GO TO 136
 135       JJ=I
 136       DO 140 I=1,JJ
           IX=(5.*AP(I)+.5)+1
           IF(IX.LE.0)  IX=1
           IF(IX.GT.110)  IX=110
           X(IX)=STAR
C
```

```
      IXX=(5.*P(I)+.5)+1
      IF(IXX.LE.0)  IXX=1
      IF(IXX.GT.110)  IXX=110
      X(IXX)=PLUS
      WRITE(22,930) T(I),(X(J),J=1,110),
     &  P(I),AP(I)
  930 FORMAT('0',F5.2,110A1,2F6.2)
      X(IX)=BLANK
  140 X(IXX)=BLANK
      GO TO 105
 9999 WRITE(22,945)  (I,I=0,22)
      CALL EXIT
      END
```